Republic of Iraq
Ministry of Higher Education and Scientific Research
University of Babylon
College of Science for Women
Department of Computer Science

Classification of Spam Emails Using Min-Hash and Deep-Learning

A Thesis
Submitted to the Council of the College of Science for Women at University of Babylon in Partial Fulfillment of the Requirements for the Degree of Master in Science\Computer Science

By
Nuha Hussein Marza

Supervised by
Prof. Dr. Hussian .Atyaa.Laftaa **Asst.prof.Dr. Mehdi Ebady Mana**

2022 D.C 1443 A.H

بسم الله الرحمن الرحيم

يرفع الله الذين آمنوا منكم والذين أوتوا العلم درجات والله بما تعملون خبير

صدق الله العظيم

سورة المجادلة . آية 11

The Head of the Department Certification

In view of the available recommendations, I forward the thesis entitled "**Classification of Spam Emails Using Min-Hash and Deep-Learning**" for debate by the examination committee..

Signature:
Name: Dr. Farah Al-Shareefi
Date: / / 2022
Address: University of Babylon/College of Science for Women

Declaration

I hereby declare that this thesis entitled "**Classification of Spam Emails Using Min-Hash and Deep-Learning** ", submitted to the University of Babylon in partial fulfillment of requirements for the degree of Master college of science for women\department of computer science , has not been submitted as an exercise for a similar degree at any other University. I also certify that this work described here is entirely my own except for experts and summaries whose source is appropriately cited in the references.

Dedication

I dedicate this thesis
To the soul of my father
To the fountain of patience my mother
To my husband
to my son
To my sisters
To my supervisors
To my family
To my friends

The researcher

Acknowledgements

In the name of Allah who most gracious and merciful. Before anything, I would like to introduce the greatest praise and thanks to the source of tender, Allah almighty for supporting me to overcome all the challenges and difficulties which I faced at my work and finish this work.

I would like to thank my supervisors Prof. Dr. Hussain.A.Lafta and Asst.Dr. Mehdi Ebady Manaa, for the encouragements and advice they provided throughout my studying period. I have been quite lucky to have a supervisors who cared so much about my work, and who responded to my questions and queries so immediately. I would also like to thank all the members of staff at Babylon University who helped me.

I must express my gratitude to those who made me appreciate the value of the science, my father - God bless his soul, my family (mother ,my husband , and sisters) for their patience and support. Their encouragement has been my greatest strength which enabled me to complete my research .I would like to thank my colleagues for, who gave me their help or advice

The researcher

Table of Contents

List of Figures		
Fig. No.		**Page**
Figure 2.1	CIA Security Criteria	14
Figure 2.2	Data Mining Process In Steps	18
Figure 2.3	An Overview of Steps Behind KDD Process	19
Figure 2.4	Jaccard Similarity of Two Sets	22
Figure 2.5	Deep Neural Network	27
Figure 2.6	Famous Model of Deep Learning	29
Figure 2.7	Kinds of Perceptron Neural Network	31
Figure 2.8	Single Node Neural Network	32
Figure 2.9	Deep Neural Network(DNN)Architecture	33
Figure 2.10	The Operation of Computing The Gradient	35
Figure 2.11	The Dropout Process	36
Figure 2.12	ReLU Plotted Function	38
Figure 2.13	Soft-max Plotted Function	39
Figure 2.14	The Confusion Matrix	44
Figure 3.1	The Proposed System of Min-Hash Deep Learning	48
Figure 3.2	Sample of The Data Set	47
Figure 3.3	The Architectures of Proposed DNN For	58

List of Figures		
Fig. No.		**Page**
	Classify Model	
Figure 4.1	Dataset Before Preprocessing	65
Figure 4.2	Dataset After Preprocessing	65

List of Tables

List of Algorithms		
Alg. No		**Page**
Algorithm 1	k-shingle algorithm	50
Algorithm 2	Min-hash hashing algorithm	53
Algorithms 3	Back propagation algorithm	55
Algorithms 4	DNN model training	60

List of Abbreviations	
AI	Artificial Intelligence
ANN	Artificial Neural Network
ARM	Association rule mining
BP	Back Propagation
CIA	Central Intelligence Agency
CNN	Convolutional Neural Networks
DBN	Deep Belief Network
DL	Deep Learning
DM	Data Mining
DNN	Deep Neural Network
DT	Decision Trees
IS	Information System
KDD	Knowledge Discovered in Databased
KNN	K-Nearest Neighbour
ML	Machine Learning
MSE	Mean Square Error Loss Function
NB	Naïve Bayes
NLP	Natural Language Processing
NN	Neural Networks
RBM	Restricted Boltzmann Machine

List of Abbreviations	
ReLU	Rectified Linear
RF	Random Forest
RL	Reinforcement Learning
RVM	Relevance Vector Machine
SL	Supervised Learning
SVM	Support Vector Machines
TED	Testing Dataset
TF–IDF	Term Frequency–Inverse Document Frequency
TRD	Training Dataset
UBE	Unsolicited Bulk Email
UCF	Unsolicited Commercial Email

ABSTRACT

The internet has become an integral part of modern life. One of the most critical aspects of the internet is collaboration. Email is a communication tool that can be used for both personal and professional(official) purposes. Spam messages are not intended to be received by addressee of emails, and therefore are often regarded as unwanted bulk emails. Every day, a wide range of people uses emails to connect globally. Being in large quantities already causes real frustration for both internet users and providers.

For instance, it degrades user analysis data, encourages network virus migration, expands stack on arrangement movement, absorbs mail server storage, wastes time and network bandwidth, and depletes the vitality of real emails among the spam.

It is therefore necessary to prevent the spread of spam. Given the fact that there are several data mining techniques beneficial in preserving security, they can also be of use in classifying spam email.

As for the present work, the min-hash technique is combined with the Deep Neural Network (DNN) algorithm to classify emails into spam and ham.

Min-hash is based primary on two main concepts are Jaccard similarity and K-shingle. Where more than example has been used for the divisions of the K-shingle as (k =3 ,4 ,5), when comparison the result.

Throughout this work, a five-layer system of hidden layers was proposed. Starting with 64 nodes in the first secret layer and 128 nodes in the second, there were 128 nodes in the third layer, followed by 64 and 64 nodes in the fourth and fifth layers, respectively. 64 batches were set up after the training set was trained. The Python environment was used to

carry out this work, and 70% to 30% of the data was taken for the purpose of training and testing the results.

The results show that a remarkably high accuracy rate (98.5%) is obtained by using combination(k=5), which means that it is an effective method to be adopted and further developed in the field of spam detection and classification.

Chapter One

Introduction

1.1 Introduction

The quantity of electronic text content accessible is constantly expanding, including electronic publications, digital libraries, electronic books, email messages, news stories, and Web sites. The automatic classification or regulation of document is necessary due to the significant rise of electronic document. Document categorization is the process of allocating a document to one or more preset classifications based on the content (text) of the document. The demand for tools to help individuals locate, filter, and manage these resources is increasing. As a result, the automated categorization of text document collections is an important topic in academia. To improve automated text data organizing, a number of machine learning approaches has been proposed. Unsupervised (document clustering) and supervised (document classification) approaches are the two primary types of these techniques [1].

The supervised learning job of email categorization topic spotting knowledge engineering was the most prevalent technique to email categorization until this machine learning approach. Expert knowledge is used in knowledge engineering to establish a set of manual criteria for categorizing emails into pre-defined categories. Using machine learning for email classification saves time and money in terms of doing away with expert personnel while maintaining accuracy [2].

K-shingle approach is one of the methods that been widely implemented to convert large set of data by grouping this data into small groups. Min-hash is the main approach in this work it employs many hash functions to generate characteristic matrix from the documents after implementing k-shingle. Then, this matrix will change to signature matrix to get the similarity of email [3].

1.2 Problem Statement

Emails are used all over world. However, many emails are spam giving rise to various problems when dealing with them. The most important of which is their huge number arriving in inboxes. So, it is difficult to distinguish and detect spam emails. Email has evolved into one of the most convenient and cost-effective methods of communication. However, the ubiquity of email led in an increase of spam emails in recent years. Emails are an efficient method of online communication due to their resources conservation and communication time reduction, making them a popular choice for private and technical communication [4].

Spam includes malware in scripts or other executable files and those may harm the user's computer. Some of the disadvantages of spam messages include: lower productivity, mail box space reduction; extension of viruses, Trojans, and resources containing potentially damaging information for certain users; compromise mail servers' stability, and consequently, leading users to spend time sorting incoming mail and deleting unsolicited ones. In conclusion: spam is not only a nuisance, but also threat to businesses due to the tremendous number spam email received by users [5].

To address this problem, a spam categorization system will be developed that can distinguish between spam and non-spam messages. In our suggested system for spam identification, a number of methods will be employed, including Min-hash techniques to find signature matrix and deep learning categorization.

1.3 Research Objectives

The objective of the proposed work is to classify the emails received by implementing Min-hash technology and deep learning techniques. Objectives can be divided into the following:

1. To implement a hybrid method from min-hash and deep learning .
2. To find out a good performance information of accuracy.

1.4 Contribution of the research

This work comes to play an important rule to find emails classifications and emails using:

1. Min-hash technique has been conducted for large dataset.
2. Using data mining classification is considered a new technique with Min-hash for emails classification.
3. The important contribution in this thesis is to present a general model for classify email for spam or ham and can be used as a personal system or a system for organization.

1.5 Information Systems (IS)

Information systems (IS) are vulnerable to a variety of security attacks that can result in severe financial losses and damage to system resources. Security risks create many forms of harm, such as database integrity security breaches, physical destruction of complete information systems facilities due to fire or water, and so on. Every occurrence that can result in information confidentiality, integrity, and availability breaches, or any other kind of information system resource harm, is referred to as a security threat [6].

With the rapid advancement of technology, we are now able to acquire huge volumes of various sorts of data. Network security is “the process of taking

physical and software preventative measures to protect the underlying networking infrastructure from unauthorized access, misuse, malfunction, modification, destruction, or improper disclosure, resulting in a secure platform for computers, users, and programs to perform their permitted critical functions within a secure environment ". Information security is inextricably linked to network security. It is critical to protect data traveling through networks and computers [7].

The goal of information security is to secure an organization's information that assets against illegal access, disclosure, interference, or destruction [8][9].

1.6 Email Classification

Email has evolved into one of the most convenient and cost-effective methods of communication. However, the ubiquity of email has resulted in an increase of spam emails in recent years. Emails are efficient method of online communication since they conserve resources and reduce communication time, making them a popular choice for private and technical communication.

Text classification of an email message is one example of supervised learning, in which the goal is to create a "probabilistic model" of a function that classifies emails into classes. A learning algorithm is given with a collection of already categorized, or labelled, instances in supervised learning of text in email messages, where a complete email dataset is one example of emails to be classified. This set is called the training set. The more representative the training data, the better the performance as larger samples tend to be more reflective of the authentic distribution of data as a whole [10].

Unsolicited bulk communications, or messages delivered to many recipients who did not ask for them, are the basic definition of spam. Unsolicited commercial e-mail is a popular alternative definition, based on the assumption that most unwanted correspondence is commercial. Many mailbox providers regard it as mail

that their consumers do not desire or about which they have expressed dissatisfaction. Spam mail, also called unsolicited bulk e-mail or junk mail that is sent to a group of recipients who have not requested it [11].

Spam includes some malware in scripts or other files that are executable and may harm the user's system. E-mail is an email that meets the following three criteria:

- **Anonymity**: the address and identity of the sender are hidden.
- **Mass mailing**: mail messages that are sent to a huge group of people.
- **Unsolicited**: email is not requested by recipients.

Spam makes up the vast majority of emails. Consequently, people have difficulty in understanding ham emails . As a result, a large number of approaches have been developed and made available to the public.

1.7 Data Mining (DM)

Data mining is the process of analyzing hidden data patterns from various standpoints in order to categorize it into advantageous information. Information is gathered and categorized into common areas, such as “data warehouses” for effective analysis, data mining algorithms to facilitate business decision-making, and other information requirements. The ultimate goal is costs reduction and revenue increase. Data mining is often referred to as knowledge discovery and data discovery [12].

Many techniques used in data mining are listed below. To pick up and get important data, the methods below are applied to non-essential data.

1. **Anomaly Detection:** is the mining of data that is erroneous or irrelevant. Anomaly detection identifies information that is devoid of facts.

2. **Association Rule Mining (ARM**): is a technique of distinguishing relationships among the attributes included in datasets.
3. **Clustering**: is a method that collects comparative data in a single cluster without the use of any predetermined show. It is a data collecting system that has its own model and is enthralling.
4. **Classification**: is a technique that has a preset demonstration and categorizes the data into specified groups. A prediction model is classification.
5. **Summarization:** is a method of presenting information in a precise shape for perception [13].

In this thesis, data mining classification methods are used to classify emails as spam or non-spam (ham) using deep neural network and min-hash techniques.

1.8 Related works

Naïve Bayes and SVM machine learning techniques are used in [5] by Muhammad Ali Hassan and Nhamo Mtetwa. They use different feature extraction methods together with two different supervised machine learning classifiers that are evaluated using four performance metrics on two open-source spam email datasets for spam filtering. They are highlighted the significance of correct coupling of feature extraction and classifier.

Chih-Chin Lai and Ming-Chi Tsai use spam email categorization NB, TF-IDF, K-NN, and SVM, in [14] by applying them to different parts in order to compare their performance for spam email categorization. The main objective from of this is to combine two methods (TF-IDF and NB) which achieve the most accurate categorization. It found that integrating different learning algorithms achieved a performance level of 90%.

Seongwook Youn and Dennis McLeod in [15], use the ontology specially designed to filter spam and unsolicited bulk email to expunge it out of the system. They use Neural Network NN, SVM classifier, Naïve Bayesian Classifier, and J48 classifier. The obtained results are in effect of datasets on performance in case of 1000 datasets. The accuracy are 95.80% using J48 classifier, 93.50% NN, 92.70% SVM and 97.20% Naïve Bayesian. The obtained results are in effect of feature size on performance in case of 10 features. The accuracy is 94.84% using J48 classifier, 83.60% NN, 81.91% SVM and 92.42% Naïve Bayesian.

Bo Yu and Zong-ben Xu proposed the use of Naïve Bayesian (NB), Neural Network (NN), Support Vector Machine (SVM) and relevance vector machine (RVM) for spam classification in [16]. Different training set size and extracted feature size are the parameters for the experiments. Experimental results showed that (NB 93.1% ,NN 84.2% ,SVM96.1% ,RVM 96.5%), as a spam rejection tool, NN classifier is unsuitable for use on its own.

Aman Kumar used ID3, J48, Simple CART and Alternating Decision Tree in [17] to classify spam email dataset. Classification accuracy is used as the basis of comparison between the four algorithms. ID3, CART and ADtree are outperformed by J48 classifier in terms of classification accuracy where ID3 0(89.11 %) J48 (92.7624%) Simple CART (92.632%) ADTree (90.915%).

Random Project method and Random Boost method are used in [18] by Dave DeBarr and Harry Wechsler. The main objective is to compare performance of small-number-of-examples-trained robust and efficient spam detection filters. TREC and CEAS are used as challenging spam application domains. Results showed that Random Boost method, in comparison to Logit Boost algorithm, dramatically improves the performance of the spam filter.

A comprehensive survey is presented in [19] by Charu C. Aggarwal and Chengxiang Zhai. They shed light on a wide spectrum of text classification algorithms, such as the Support Vector Machine, Decision Tree, and Rule-based Classifiers. Recent studies show that, to dramatically improve the quality of the underlying result, it is recommended to incorporate linkage information into the classification process.

In [20], Shikhar Seth and Sagar Biswas use deep neural network to classify a email into spam or not-spam (ham). The whole content, i.e., image and text, is analyzed by processing it through independent classifiers using convolutional neural networks. By forging the image and text classifiers, they suggested two hybrid multi-modal architectures. Their experimental results outperformed current state of the art methods and provided a new starting point for future research in the field.

In [21], et al. used artificial neural network to predict whether an email is spam or not. When the model is tested, the general result is 85.31%. This study demonstrates the feasibility of artificial neural network for classification of emails.

M. Bassiouni, M. Ali, and E. A. El-Dahshan in [22] studied a 10-fold cross validation to provide the accuracy by forest technique, in comparison to other classifiers. They use RF, ANN, Logistic Regression, SVM, Random Tree, KNN, Decision Table, Bayes Net, NB, and RBF. The accuracies are 95.4, 92.4, 92.4, 91.8, 91.5, 90.7, 90.3, 89.8, 89.8, and 82.6, respectively. The best performance is Random Forest performed best with an accuracy of 95.45%.

To find out the classification accuracy, S. Sumathi and Ganesh Kumar Pugalendhi in [23] introduced Random Forest integrated with Deep Neural Network. A preordained probability of attributes is used by Random Forest algorithm in constructing their decision trees. To rank important features, Gini

measure is examined. The main objectives are: 1) to grade the features using RF algorithm, and 2) to train data using Deep Neural Network Classifier. Backpropagation algorithm in batch learning mode is used to train Deep Neural Network Classifier model (DNNs). This requires the entire training data to learn at once. Dynamically fitting the detector process to the new data patterns is used until it reaches spam coverage. Results showed that KNN and Support Vector Machine (SVM) classification rate is less than that of DNN. The latter scored an accuracy of 88.59% taking into consideration the five top ranked features.

Table 1.1 : The Summarization of the Related Works.

N0.	Ref	Author name	Method	Evaluation
1	[14]	Chih-Chin Lai , Ming-Chi Tsai 2005	NB, TF-IDF, K-NN, and SVM	-
2	[15]	Seongwook Youn , Dennis McLeod 2007	Naïve Bayesian,SVM,NN,J48	NN 83.6% SVM 81.91% NB 92.42% J48 94.84%
3	[16]	Bo Yu , Zong-ben Xu 2008	NB ,NN,SVM ,RVM	NB 93.1% NN 84.2% SVM96.1% RVM 96.5%
4	[17]	Aman Kumar , Suruchi Sahni	ID3, J48, Simple CART and ADTree	ID3 0(89.11 %) J48 (92.7624%)

		2011		Simple CART (92.632%) ADTree (90.915%)
5	[18]	Dave DeBarr , Harry Wechsler 2012	Random Boost algorithm Logit Boost algorithm	-
6	[19]	Charu C. Aggarwal , ChengXiang Zhai 2013	decision trees, rules, Bayes methods, nearest neighbor classifiers, SVM classifiers, and neural networks	-
7	[20]	Shikhar Seth , Sagar Biswas 2017	Text CNN	97.54%
8	[5]	Muhammad Ali Hassan , Nhamo Mtetwa 2018	SVM , Naïve Bayes	93%,99%(SVM) 80%,98% (NB)
9	[21]	Ahmed Alghoul , Sara Al Ajrami, Ghada Al Jarousha, Ghayda Harb, Samy S. Abu-Naser 2018	ANN	85.31%
10	[22]	M. Bassiounia,M.Alib, and E. A. El-Dahshan 2018	RF, ANN, Logistic Regression, SVM, RandomTree, KNN, Decision Table, Bayes Net, NB, and RBF.	RF(95.4), ANN(92.4), Logistic Regression(92.4), SVM(91.8), RandomTree(91.5)

				KNN(90.7), Decision Table(90.3), Bayes Net(89.8) , NB(89.8), and RBF(82.6)
11	[23]	S. Sumathi1 · Ganesh Kumar Pugalendhi 2020	SVM,KNN,DNN	DNN 88.59%

1.9 Thesis Structure

This thesis begins with general introduction to some concepts used and the contents of the other four chapters are as follows:

Chapter Two "Theoretical Background "introduces an introduction to method and technique. Additionally, proposing how these technique work and its result. Furthermore, it explains the evaluation method that has been used in this thesis.

Chapter Three "**The Proposed System Methodology**" introduces the proposed system, explained every method used and how its implement in project. It explains how gate k-shingles is similar to emails with little time.

Chapter Four "**Results and Discussion**" introduces the results of the proposed system with evaluation methods in addition to the implementation results.

Chapter Five "**Conclusions and Future Works**" presents conclusions and suggestions for future work.

Chapter Two

Theoretical Background

2.1 Introduction

Nowadays, with the fast progress in the technology we are capable to collect massive amounts of different types of data. Document similarity measure has direct impact to document-based classification. The majority of email are unstructured and also not well organized. Thus, users faced difficulties to find spam email . Thus, to many techniques have been generated and present to the world, what we have used to participate in this thesis have been proposed in this chapter.

Data mining is a method utilized to extract insightful and interesting data from databases. Data mining is very useful technique generated to help people focusing on the most important information [24] . Therefore, it can be used for many applications such as market analysis, customer reservation, cheat detection, science investigation and so on. basically, data mining has so many efficient techniques been: classification, clustering and association rule. in this thesis we typically employ one of the most effective common technique which is classification [25].

2.2 Information Security (IS)

Three information security properties make up the information security model. The desired goals are: confidentiality, integrity, and availability, or CIA, an acronym used to help keep these principles in mind. The definition presents three key objectives essential to computer security:

- **Confidentiality** guarantees that private or confidential information is not made available or disclosed to unauthorized individual.
- **Integrity** guarantees that information and programs are changed only in a specified and authorized manner.

- **Availability** guarantees that systems work promptly and service is not denied to authorized users [26] [27].

These three concepts form what is often referred to as the CIA triad as **Figure (**2.1)**.**

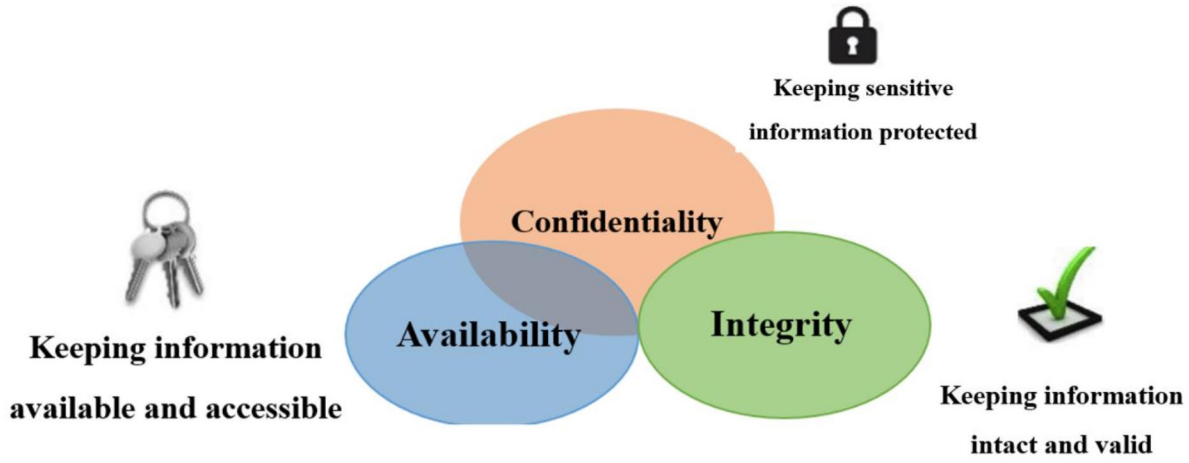

Figure 2.1: CIA security criteria [28]

An access control refers to a set of security protocols used to prevent unauthorized access to a computer or a network. The Access Control List (ACL) is used to specify when a person has specific access capabilities on a variety of devices. If you cannot get onto the business network, then you do not have authorization to use the high-speed color printer. Authentication confirms a user's identity in which case an individual or user must present one of the following: (1) a password, (2) a token or card (i.e., a badge), or (3) a biometric, like a fingerprint. Authentication is related to access control, which is concerned with a subject's (person or computer system process) capacity to interact with an item (like a file or a hardware equipment). If you want to withdraw money from an ATM, you will need your bank card, which is something you own and for which you will need to

know the PIN. Multifactor authentication requires more than one kind of authentication. Passwords are the most often used type of authentication [27] .

2.3 Spam Emails

Spam, also known as Unsolicited Commercial Email (UCE) or unsolicited bulk email (UBE), is ubiquitous in email communication. Spam email, or unwanted email messages that fill our inboxes, is a problem for Internet users, companies, and politicians. According to some estimates, over 100 billion spam messages are sent/received every day, accounting for up to 85% of worldwide daily email traffic. Spam is a costly problem that, according to many experts, is only growing worse. Spam is unlikely to go away very soon because of its economics and the challenges in preventing it [29].

The increased usage of email has resulted in a massive amount of data being generated and exchanged. While the ability to quickly exchange information attracts individuals and businesses, it also draws those who send unwanted and unsolicited communications via the internet. Spam is the term for this sort of communication [30].

Generally, spam is commercial, fraud, or insulting communications intended to take advantage of the recipients. Spam detection began with manual message screening, then moved on to basic filtering rules that could recognize a message with certain characteristics. For many countries, spam is a continuously growing challenge on technical, economic, and security levels. As a result, addressing its issues will need a comprehensive strategy. The topic of spam, in particular, has the following problems to consider:

Spam is a costly issue for the Internet's infrastructure as well as it is consumers. Spam consumes a lot of network resources, and it is especially bad for nations with

limited Internet access and bandwidth. ISPs put forth a lot of work to control this traffic, and end users must be cautious about accepting spam that contains malware or is a hoax. Costs of receiving or unintentionally sending large amounts of spam messages may be substantial for mobile data customers and those who subscribe to metered services. There are also expenses involved with repairing systems that are infected and/or attacked by spam-enabled malware, as well as costs related to stolen user data [30].

Spam's economics, in general, is significantly skewed in favor of spammers. Spam communications are inexpensive to send. In actuality, message receivers, ISPs, infected users, and network operators bear the majority of the expenses. As new apps and methods of data transmission on the Internet become available, the nature of spam evolves. Spammers are improving their capacity to exploit such platforms to offer increasingly intrusive and destructive methods of stealing personal data, causing network damage, and infecting computers [30]. Spam impacts a wide spectrum of Internet users, and no single company can completely eliminate the problem of spam. To solve the challenge, a global, multi stakeholder community must collaborate. Spam causes a lack of trust on part of the user and is seen by some as a barrier against using Internet and e-commerce. In addition to the direct harm to users and the strain on network resources. There is also the risk of a user's reputation being tarnished if spammers steal their identity and use it to transmit spam.

Communities that implement antispam measures may face reprisal (e.g., victims of Distributed Denial of Service (DDoS) attacks, or hacking). So, it is critical that members of the global antispam community not only offer advice on how to counter spam, but also provide technical and other support in the face of retaliation [31].

The use of standard machine learning approaches to construct a spam detection model was the beginnings of automatic spam detection. Simple approaches like blacklisting and content-based machine learning algorithms are commonly employed for spam filtering in conventional spam detection. Existing spam detection algorithms worked well on lengthier email messages, but in recent years, spam detection in short and noisy platforms has posed additional challenges. Deep Learning Technologies are the most recent advancement in the field of categorization. They are well-known for their outstanding performance in Natural Language Processing (NLP) [30].

2.4 Data Mining Techniques

Data mining is the process of obtaining meaningful information from huge sets of data using statistical and artificial intelligence join methods. It is defined as a computer procedure that involves examining data in order to discover patterns and relevant information. The major goal of this technique is to find patterns that have yet to be identified. So, after these patterns have been discovered, they may use them to make choices about work appraisal. Data mining methods will be used to forecast future trends in a variety of areas, including information technology, health, sociology, and physics. Data mining, on the other hand, is also known as knowledge discovery, knowledge mining from data, and knowledge extraction [31].

The data mining technique goes through several stages, as shown in the diagram (2.2). The following explanation includes a quick description of each of the data mining steps:

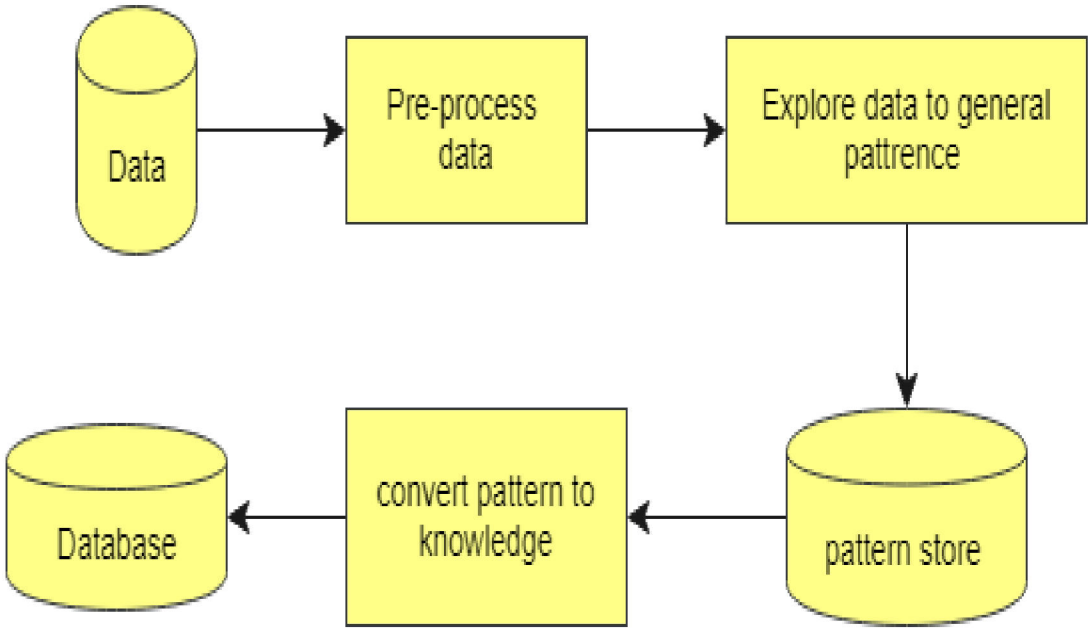

Figure 2.2: Data mining process in steps [32]

1. Extracting raw data and transforming it into data that has been pre-processed.
2. Using a multidimensional data warehouse system to manage and store data.
3. Preparing data for data modeling and using one of the approaches to perform data modeling as (Artificial Neural Networks (ANNs) and Decision Trees (DTs) , Classification , Clustering ,Rule induction etc.).
4. Finally, the information obtained from data modeling is turned into a model that may be expressed in a graph or table [32].

Some individuals use the term "data mining" interchangeably with "Knowledge Discovered in Databases" (KDD). Others, however, believe that data mining is an important stage in the knowledge discovery process [33].

Basically, knowledge discovery is a process proposed in Figure (2.3) below and it contains next steps:

A. data cleaning (to take off noise or irrelevant data),

B. data integration (whereas many sources of data may be joint),

C. data selection (only relevant data are retrieved),

D. data transformation (whereas data have transformed or integrated into suitable forms for mining),

E. data mining (a major process where smart methods are used to extract data) .

F. pattern evaluation (to distinguish the good patterns),

G. knowledge presentation (knowledge representation and visualization techniques are utilized to attend the entire knowledge to the user .

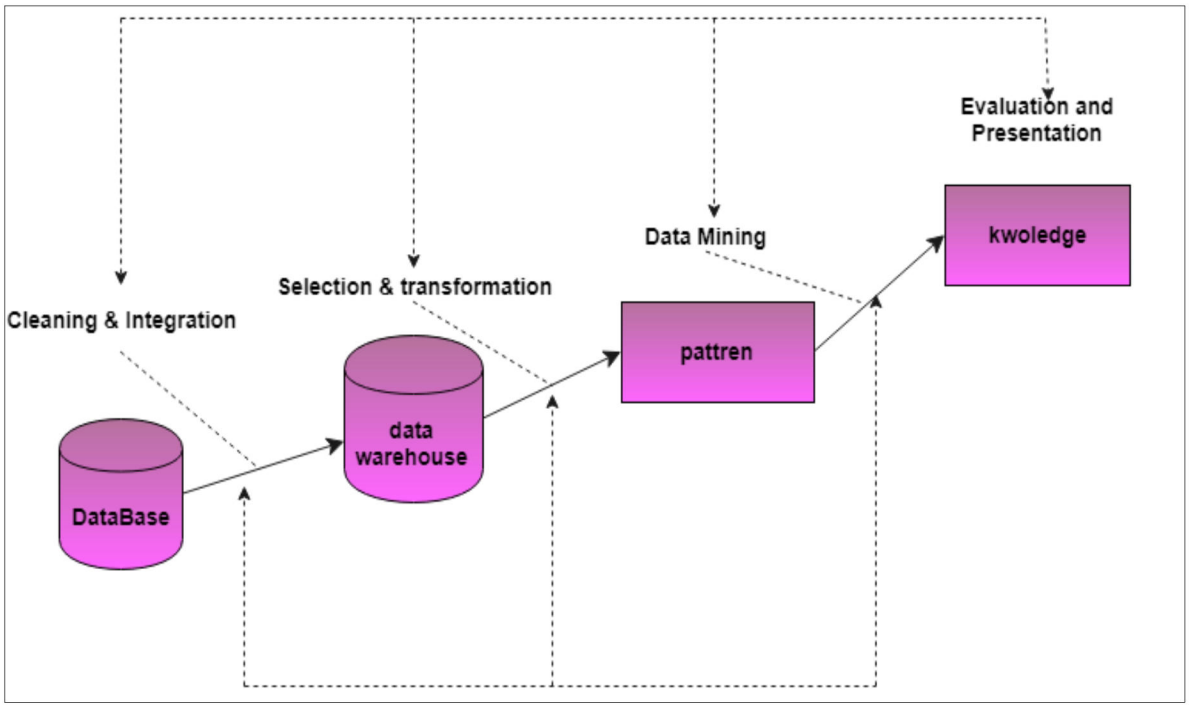

Figure 2.3: An overview of steps behind KDD process [34]

Furthermore, different algorithms and knowledge discovery techniques are used such us Classification, Clustering, Association Rules etc. [34].

- **Classification**

Classification is a common data mining approach that uses a set of pre-classified examples to create a sample that may group records in big datasets. A decision tree or neural network-based classification algorithm is used in this method. In order to check the reliability of classification rules, test data is used in classification. The classification job uses as input a set of data known as the training set. Each of these records has a set of characteristics, one of which is the record's class. This is an example of separating spam from helpful and crucial email communications. Classification is a form of supervised learning. The fundamental idea behind supervised learning algorithms that use named classes is to learn particular classes and then use classification and prediction algorithms, which are generally mathematical functions or statistical and probability models, to forecast new classes. Unlabeled classes are assigned to designate classes by the classifier [35].

The primary objective of classification is to find a suitable sample for the class attributes based on the values of the other attributes. The sample will be used to forecast the class attribute of the observed records in the future.

Take, for example, the following set of records that describe the condition of university employees. Assume that each of these records contains the following attributes: (i) the professor's name, (ii) his or her position (for example, associate professor), (iii) the number of years she or he has worked at an institution, and (iv) the class attribute, which shows whether or not the professor is eligible for tenure. Consider the following records in the collection:

Anna, Assistant Prof, 3, no,

Harry, Assistant Prof, 7, yes,

Scottb, Full Prof, 2, yes,

Typically, we can predict that the classification algorithm will identify a sample based on the prior input :

"IF position=Full Prof OR years > 3 THEN tenured=yes". Thus, given a new record Barbara, Full Prof, 4?

The sample result will predict a yes response for the missing class value [36].

2.5 Emails Similarity

One of the most efficient techniques for calculating document similarity is the Jaccard similarity coefficient. It's a measure of how similar two sets A and B are. The formal calculation is as follows (Eq.2.1) [37]:

$$J = \frac{|A \cap B|}{|A \cup B|} \qquad \textbf{2.1}$$

In theory, the Jaccard is a similarity metric that runs from 0 to 1. Where 1 denotes that the two utilized items are comparable, while 0 denotes that they are entirely unlike [38][39].

Figure (2.4) depicts the basic notion of Jaccard similarity and how it works, with A denoting the first set and B denoting the second set, and A denoting the Jaccard similarity of the two sets.

Based on the preceding equation (2.1), the Jaccard similarity can function just good for small datasets, but if we want to assess the similarity of a big number of emails, we need to use a more complex approach called Min-hash, which we will discuss in detail in section 2.6.

When we wish to measure the similarity of two sets, we usually follow a set of processes to get an accurate result. The next part provides a quick overview of each stage and why it is necessary to execute it.

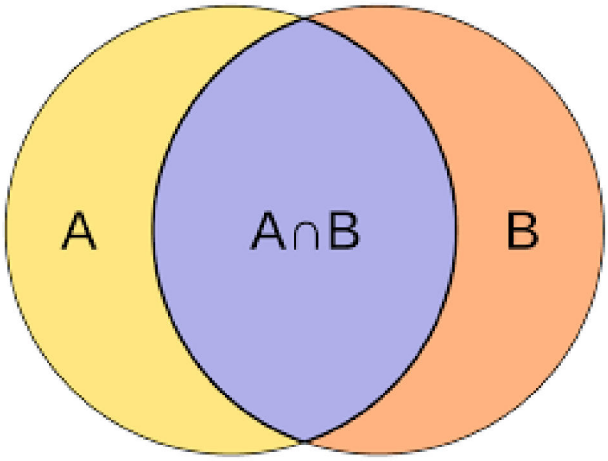

Figure 2.4: Jaccard Similarity of two sets [39]

2.6 Min-Hash Technique

In section (2.5) we have mention that Jaccard similarity works great for small datasets with tiny characteristic matrices, but that when the set of shingles of all documents is big, measuring Jaccard similarity would be a significant burden, which is why Min-hash exists. The Min-hash method generates a quick approximation of Jaccard similarity. The fundamental idea behind this method is to divide a huge number of shingles into little representations known as signatures, which will be used to compare email similarities later [40].

It's worth noting that the higher the number of signatures, the similarity finding between emails will be accurate [41] .

For email similarity estimations, Min-hash uses hash functions. The original computation of the Min-hash method is as follows (Eq.2.2):

$$\mathbf{h(x) = ax + b \bmod p} \qquad (2.2)$$

In equation (2.2), the following phrase is used: The tokens (shingles) in the original characteristic matrix are denoted by the letter **(x)** . **a** and **b** are two random numbers that are less than or equal to the prime number (**p**). This hash function

will produce distinct random **a,b** values, allowing us to construct random hash functions to select different a and b values. The number **p** is a prime number (slightly larger than the total number of shingles sets) .

Min-hash is based primary on two main concepts are Jaccard similarity and K-shingle, of which are defined in section (2.6.1 K-shingle) and discussed in further depth in chapter three. After a series of computations, the characteristic matrix becomes the signature matrix, which may then be used for Min-hash.

2.6.1 K-Shingle Technique

To transform the input emails dataset into shingles, K-shingle was used. One of the most active techniques to define the similarity of emails is to create a csv text file terms. Shingles are used as a result. This would aid in determining the degree of resemblance between emails, even if the phrases were written in a different sequence. So, after shingling, each email will provide a number of brief stages called tokens [38] . The most essential step is to choose a shingle size that will assist define the email's resemblance. Thus, selecting a number that is too little will result in a similarity result in all emails, but selecting a value that is too big will result in no similarity result. For example, if k =1, the result will only be displayed if the words or a character are utilized. If K=3 wants to use with short texts, but if k=8 and above prefers to use with long and research.

Because utilizing shingles would enhance the process's complicity, it has been recommended to treat it as a shingle itself, rather than using it directly. Hashing reduces the size of the shingles and the necessary space in general, but it is still too huge to utilize, so we produce fresh token hashes to maintain the email's resemblance. A Min-hash method may be used to accomplish this. The sample below demonstrates how shingles are produced

depending on emails. For example, suppose we have two shingles for three emails. E1= "**It's quite cloudy today**", E2= "**It's quite cloudy**", E3= "**It's quite**", the dictionary will look like **{"it's quite", "quite cloudy", "cloudy today"}**.

After shingling the emails, the next step is to create a characteristic matrix. It represents email sets for which the similarity may be determined using one of the similarity metrics. The following characteristic matrix depicts four sets of size **N** emails and size **M** shingles. The matrices will be **(M*N)**, where **N** is the number of columns and **M** denotes the number of rows [42].

Table 2.1: Characteristic Matrix of sets

Shingles	E1	E2	E3
it's quite	1	1	1
quite cloudy	1	1	0
cloudy today	1	0	0

The produced Characteristic matrix, as shown in table (2.1), is made up of shingle of tokens that are formed after shingling the emails. The number "1" denotes the presence of tokens in the set, whereas "0" denotes their absence.

For example choose (h1=0.5x+0.3 mod 7) and (h2=0.1x+0.2 mod 7),Table (2.2) shows Characteristic matrix generated from Characteristic Matrix of set in **Table** 2.1

Table 2.2: Characteristic Matrix

Shingles	E1	E2	E3	E4	H1	H2
B	0	0	1	0	3	0
E	0	0	1	0	15	14
A	1	0	0	1	10	11
D	1	0	1	1	5	10
C	0	1	0	1	0	6

2.6.2 Signatures Matrix

Signatures matrices contain the information taken from the emails. The signatures we are thinking of using to build the sets are made up of the results of massive calculations, each of which is a Min-hash of the characteristic matrix. Each Min-hash signature that we propose to produce is a change in the characteristic matrix's rows and columns. To Min-hash a collection using its characteristic matrix, we must first permute the values in the columns and obtain the value of the first row in which the column has a value of 1. The hash in an email is demonstrated in the following example. Let us suppose we have two hashes (2) and the k-shingle values of emails as given in table (2.2). Using (Eq.2.3), we can construct two hashes as shown below.

$$h\pi(r) = \min \pi(r) \qquad (2.3)$$

Where π is randomly permutation and **r** the number of the first (in the permuted order) row in which column r has value 1 [41].

Typically, the signature matrix will be as follows:

Table 2.3 : The Signatures matrix

#	E1	E2	E3	E4
H1	5	0	3	0
H2	10	6	0	6

2.7 Deep Learning Techniques

Deep learning (DL) is a machine learning (ML) and artificial intelligence (AI) approach that mimics how individuals learn [43].

It has a hierarchical architecture in which each higher layer builds on the preceding lower layer, giving it the name hierarchical learning for the first time [44].

Deep learning is a crucial component of data science, which includes statistics and predictive modeling. It is especially beneficial for data scientists who must acquire, analyze, and comprehend large amounts of data; deep learning makes this feasible. Processing and comprehending massive amounts of data; deep learning speed up and simplifies this process. Deep learning has the advantage of creating the feature set on its own, without the need for human interaction. Unsupervised learning is not only quicker, but also more accurate in most cases.

Deep Learning algorithms rely on neural networks in the same way that the human brain uses millions of neurons to compute information.

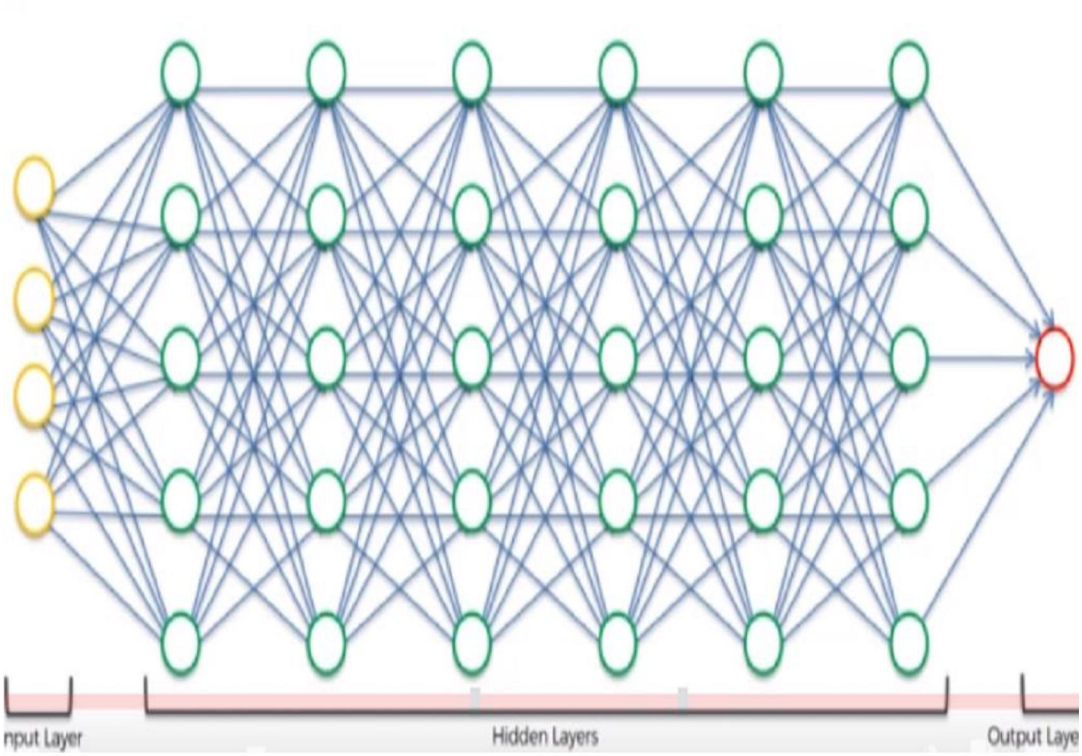

Figure 2.5: Deep Neural network [45]

Let's discuss layers' type:

1. Input layer – The input layer has input features a dataset that is known to us.
2. Hidden Layer – Hidden layer, just like we need to train the brain through hidden neurons.
3. Output layer – value that we want to classify [45].

2.7.1 Basic usages of deep learning

Deep learning is used to solve a diversity of issues in computer vision applications, including object recognition, object detection, segmentation, text classification, image classification, image caption, speech recognition, generative models, manufacturing, biometrics recognition system, similarity learning, gaming, and many more [46].

2.7.2 Deep learning models

Supervised deep learning, unsupervised deep learning, and semi-supervised deep learning are the three primary types of deep learning models, each having its own network topologies and applications.

• Supervised Deep Learning

The model is trained using labeled data in Supervised Deep Learning techniques. It is then adjusted using the learning method, and during testing, the model should predict the correct answer without depending on any labels. The two primary domains of supervised deep learning are classification and regression issues. A convolution neural network is one of the most often used supervised models (CNN) [47].

•Unsupervised Deep Learning

The model is trained on unlabeled data in unsupervised Deep Learning methods, and the model tries to uncover patterns and features on its own. Restricted Boltzmann Machine (RBM), Deep Belief Network (DBN) are examples of unsupervised deep learning architectures.

•Semi-Supervised Deep Learning

Semi-Supervised Deep Learning is a level of learning that is halfway between supervised and unsupervised. To support a huge quantity of unlabeled data, a little amount of classified data is required. When extracting important data characteristics is challenging and labeling samples take a long time, this technique comes in handy. General advection is a popular method that employs this strategy. General adversarial networks are a typical technique that uses this strategy (GANs)[48].

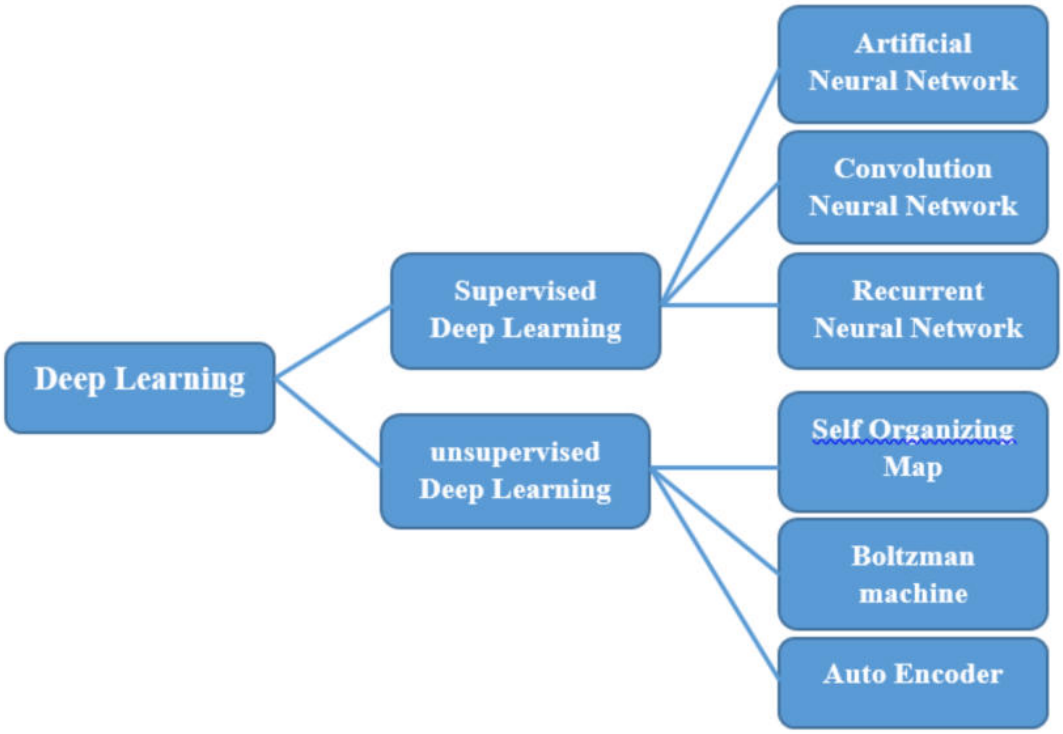

Figure 2.6: Famous Model of Deep Learning

2.8 Deep Learning (DL) in Artificial Neural Networks (ANNs).

A typical Neural Network (NN) consists of several basic, linked processors, referred to as neurons, each of which produces a sequence of real-valued activations. Sensors detecting the environment activate input neurons, whereas weighted connections from previously active neurons stimulate additional neurons.

By initiating activities, certain neurons may be able to impact the surroundings. The goal of learning or credit assignment is to find the weights that cause the NN to display desirable behavior, like driving a vehicle. Such behavior may need extensive causal chains of computing stages, where each step alters (often in a non-linear way) the cumulative activation of the network, depending on the issue and how the neurons are linked. Deep Learning is concerned with properly allocating credit across a wide range of stages.

For decades, if not millennia, shallow NN-like models with a few such phases have been around. Models with many nonlinear layers of neurons have been around since the 1960s and 1970s. Backpropagation (BP) was developed in the 1960s and 1970s for teacher-based Supervised Learning (SL) in discrete, differentiable networks of arbitrary depth, and it was first used to NNs in 1981. However, by the late 1980s, BP-based training of deep NNs with many layers is shown to be challenging in reality. By the early 1990s, it had become an explicit study topic. With the aid of Unsupervised Learning, DL became more practical to some level (UL). In the 1990s and 2000s, solely supervised DL also experienced significant advancements. Deep NNs have finally gained widespread attention in the new century, owing to their superior performance in several essential applications over other machine learning approaches like as kernel machines. Indeed, supervised deep NNs have won several official international pattern recognition competitions since 2009, producing the first superhuman visual pattern recognition performances in restricted domains. Deep NNs have also become significant in the broader subject of Reinforcement Learning (RL), where no supervising teacher is present. Feedforward (acyclic) and recurrent (cyclic) NNs (RNNs) have both won competitions [49].

DNN stands for deep neural network, which is based on a feed-forward neural network. The backpropagation method is used to teach deep learning using stochastic gradient descent. There are several hidden layers in the deep learning network. These layers are made up of tanh, max out activation, and rectifier neurons. The global model is computed throughout the network using the multithreading approach. The deep learning approach is used to abstract the data. Deep learning is used to improve text analysis and classification accuracy [50].

2.8.1 Deep Neural networks

Artificial neural networks (ANNs) are statistical models that replicate biological neural systems in the actual world with the objective of processing data similarly to the human brain [43] [51][52].

Multiple nodes make up an ANN, and these nodes represent biological neurons. The nodes are connected by connections, which respond to one another and have weight. These nodes accept data as input, perform simple operations on it, and then transfer the results of those operations to other neurons. By altering the weights' values, the ANN may learn. Neuron characteristics, ANN topology, and learning (training) techniques all influence the features and behaviors of ANNs.

A Perceptron neural network is the earliest type of artificial neural network, and there are two types of Perceptrons: single layer Perceptrons and multilayer Perceptrons, often known as shallow neural networks.[53].

Single-layer neural networks are used to learn patterns that can be separated linearly, whereas multilayer neural networks are used to learn patterns that cannot be separated linearly.

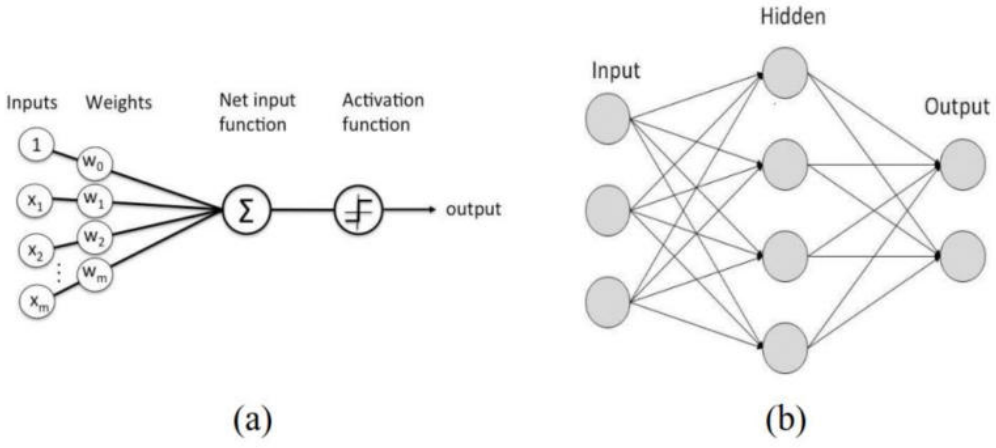

Figure 2.7 : Kinds of Perceptron Neural network **(a)** Single layer, **(b)** Multilayer [54][47]

Figure (2.8) depicts the components of a single node neural network, which comprise input vectors (X1,X2,.....,Xm), weights vectors (W1,W2,....Wm), input function (Net), activation function, and loss function (error function)[55]**.**

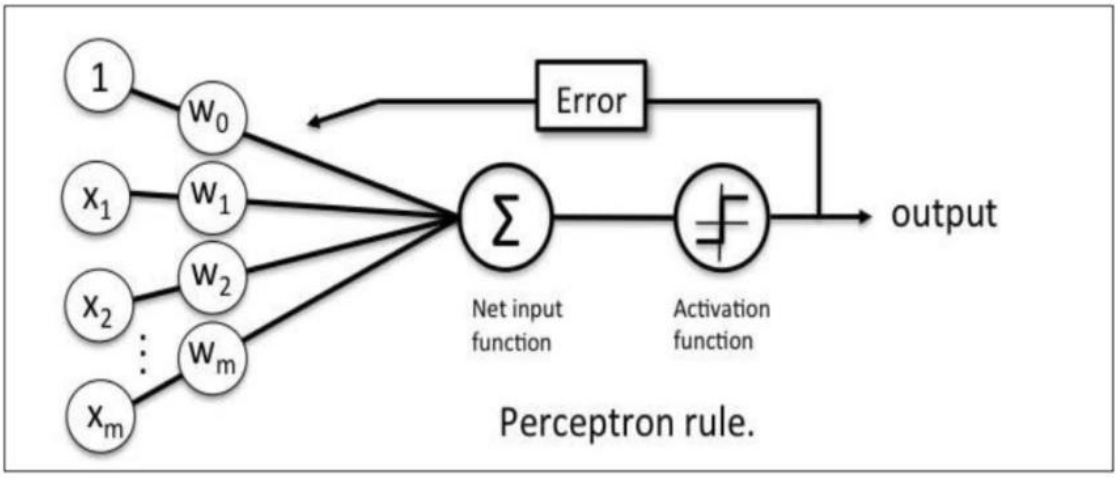

Figure 2.8: Single node neural network [55]

The activation function f of a neuron's activity to create output may be defined as follows:

$$\boldsymbol{y = f(Net)} \qquad 2.4$$

Where Net is the neuron's cumulative input stimuli (input function) and it is a nonlinear function (activation function) of the net, as defined by equation (2.5):

$$\boldsymbol{Net = x1\ w1 + x2\ w2 + \cdots + xm\ wm} \qquad 2.5$$

A loss function is a tool that evaluates a neural network's performance on specific data using the network's output. The loss function will provide a larger value if the ANN output forecast is fully disturbed. A lower number will be produced if the output projection is very excellent. The network weights will be modified based on the loss function's output value. A basic problem may be solved with a shallow neural network (maximum two layers). When a complicated pattern, such as a visual pattern, needs to be identified, several hidden layers are

necessary, and the network is referred to as a Deep Neural Network (DNN) [43][48].

A Deep Neural Network (DNN) is a shallow neural network with more than two hidden layers. Each neuron in each hidden layer performs a specific job, and each neuron in each hidden layer identifies a shape that is more complicated than the shape recognized by the neuron in the previous hidden layer. In this situation, a Deep Neural Network is used, which is more potent than a shallow network at overcoming pattern recognition [56]. **Figure** (2.9) shows how a DNN works.

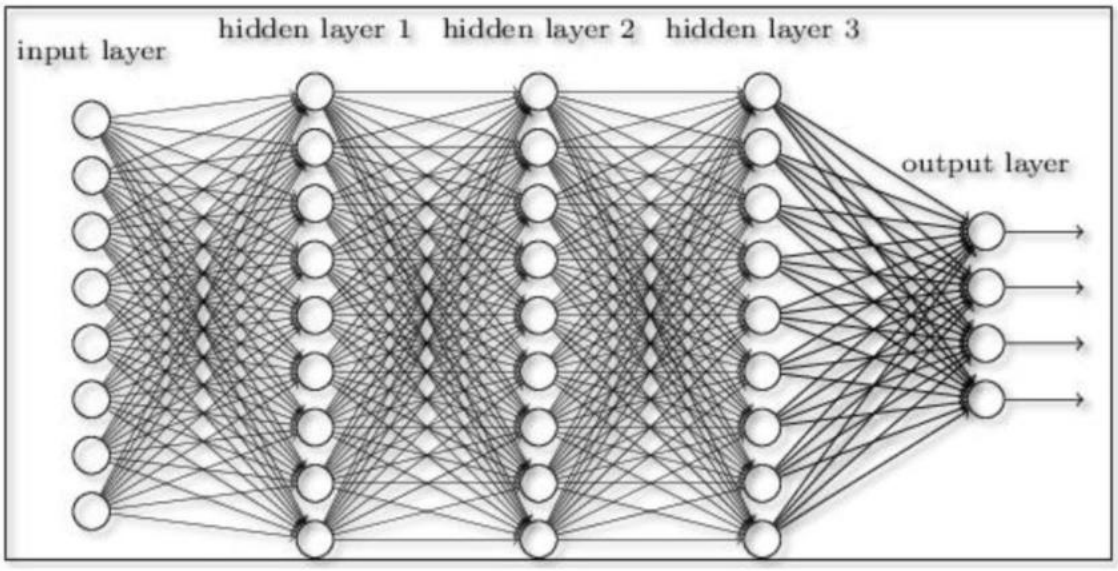

Figure 2.9 :Deep neural network (DNN) architecture [57]

2.8.2 Multi-Layer Neural Networks (Multilayer Perceptron MLP)

The MLP is a feed-forward artificial neural network-based classifier [58].

The sample does not need to be saved by MLP. A multi-layer neural network which is a type of supervised learning network that uses a succession of layers, each of which combines an affine operation and a non-linearity, to generate a prediction or classification [59].

It is made up of many layers of nodes, each of which is fully linked to the next. The input layer's nodes represent the data that is being entered. All other nodes use a linear combination of inputs with the node's weights and an activation function to map inputs to outputs. Backpropagation is a popular MLP training method. As indicated in equation, the activation function is computed (2.6& 2.7). Tanh is a number that spans from 1 to -1, and y^i is the i^{th} neuron's output .

$$x(y_i) = \tanh(y_i) \qquad 2.6$$

$$y_i = b^k + w^k * y^{k-1} \qquad 2.7$$

with parameters b^k (a vector of offsets) and W^k (a matrix of weights).

2.8.3 Backpropagation and Optimization Function

Back-Propagation (BP) algorithm, which relies on weights updating method, is a 36 common algorithm used to train a multiple-layer network on a given pattern. By changing starting weights, the back-propagation method tries to minimize the resultant loss error between the real and predicted output [49]. This is accomplished by propagating the mistake back to the previous layer. A function called Optimization Function is used to adjust the weights of a neural network. During the training phase, optimization functions adjust the network's weights in order to achieve the lowest loss function value, to make the model's output as accurate as feasible. By updating the weights network according to the loss function's output, the optimizer links the loss function with network parameters. The gradient, i.e. the partial loss function derivative concerning weights, is generally assessed using optimization algorithms, and the weights are changed in the opposite direction of the measured gradient [49] . This cycle is repeated until the model's function is reduced to a minimum. **Figure** (2.10) depicts the process of determining the gradient as well as the loss function's connection to the network weights.

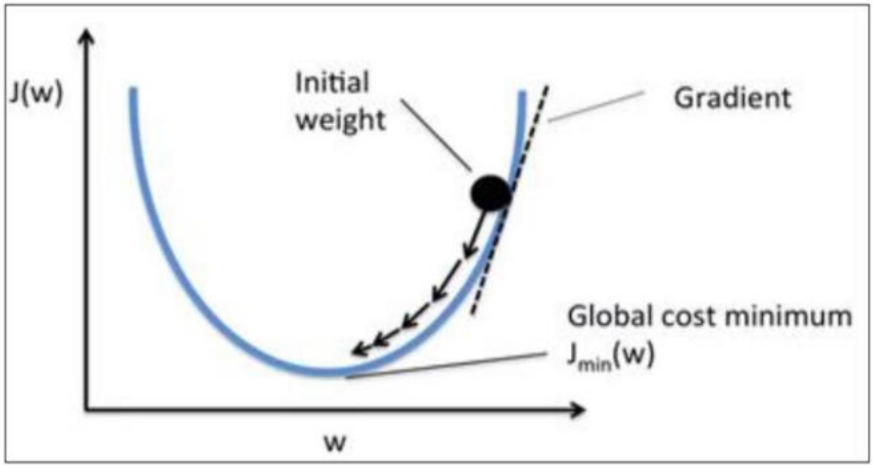

Figure 2.10: The operation of computing the gradient [60]

2.8.4 Regularization of deep learning

One of the most difficult problems in machine learning is creating an algorithm that performs well not only on training data but also on new inputs. Many machine learning techniques are specifically designed to decrease test error, which may come at the price of greater training error. The term "regularization" refers to all of these techniques. Regularization is defined as "any adjustment we make to a learning system with the goal of reducing generalization error but not training error" .There are several methods for regularization. Some people apply additional limitations to a machine learning model, such as parameter value limits [61].

Overfitting is one of the most serious problems that network models encounter, and it happens in both the instances of a complicated model and a bad dataset. Because overfitting makes it difficult to generalize the model to new data, the model's accuracy will be high during training and poor during testing. One of the most significant techniques for avoiding the overfitting problem is to use regularization layers. The 50 dropout layer is the most essential of the regularization layers.

During the training phase, the dropout layer picks random neurons and sets their weights to zero, reducing the connection between neurons in the network. This process occurs according to a probability value, generally not more than 50%. from the number of neurons in the layer. During the training phase, the dropout layer is a straightforward technique to effectively limit the model's susceptibility to noise [54]. The dropout process is illustrated in Figure(2.11).

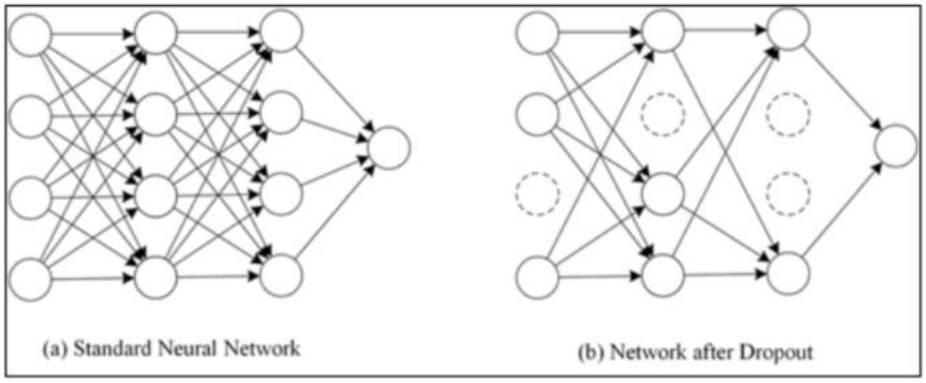

Figure 2.11: The dropout process**.**

To correctly simulate complex activities, a significant number of hidden units are required. However, such a sophisticated adaptation on training data might result in overfitting, which would prohibit good accuracy on testing data. Dropout regularization can successfully solve the overfitting problem. Dropout regularization randomly sets a specified proportion (typically half) of the activations to zero during training in the fully connected layers of feed-forward NN. As a result, concealed units that activate the same output aren't included. The vanishing gradient problem has been documented in commonly used sigmoidal units, which is typically accompanied by delayed optimization convergence to a bad local minimum. The problem is solved by the Rectified linear (ReLU) unit, which has a partial derivative of 1 when activated above 0. The ReLU function can be defined as follows:

$$h_i = \max(w_i^T x, 0) = \begin{cases} w_i^T x & \text{if } w_i^T x > 0 \\ 0 & \text{otherwise} \end{cases}, \quad 2.8$$

where w_i is the weight vector of the i^{-th} hidden unit, and x is the input vector. The ReLU function is therefore one-sided and does not enforce a sign symmetry or anti-symmetry. The primary drawback of utilizing the ReLU, on the other hand, is that it allows a NN to easily get sparse representation. It also results in a less costly calculation since the exponential function in activations is not required, and sparsity may be taken use of.

2.8.5 Activation Function

The activation functions are used to obtain the neural network's output. Activation functions come in a variety of shapes and sizes, depending on the task at hand. In general, there are two types of activation functions: linear and non-linear activation functions. Only forward propagation neural networks employ the linear activation function, and to solve simple problems that can be represented linearly. Linear activation functions have a number of drawbacks, one of which is that they cannot be utilized in back-propagation networks since the function's derivative is constant. A non-linearity is a significant aspect that aims to get it in a multilayer neural network to make it non-linear. The significance of a non-linear activation function in a neural network may be attributed to the availability of data that cannot be segregated in a linear fashion. As a result, when just a linear activation function is utilized, a multi-layer or deep neural network does not benefit from the extra layer. For some networks, the employment of a nonlinear activation function is critical, this due to its ability to map the output of network into limited range. Sigmoid, **tanh**, **Soft-max**, and **ReLU** are the most prominent non-linear activation functions used in neural networks [49][53][62].

a. Rectified Linear Units (ReLU)

The ReLU activation function is a nonlinear activation function that is commonly employed in deep neural networks [49].

If given a negative value, this function returns zero, but if given a positive value, it returns the same value. In another notion, ReLU compares the input value against zero and chooses the highest value as the winner. Equation (2.9) clarifies the ReLU activation function, which is displayed in Figure (2.12).

$$\boldsymbol{ReLU}(\boldsymbol{x}) = \boldsymbol{max}(\boldsymbol{x}) \qquad 2.9$$

Where **x**: is the neuron‘s input.

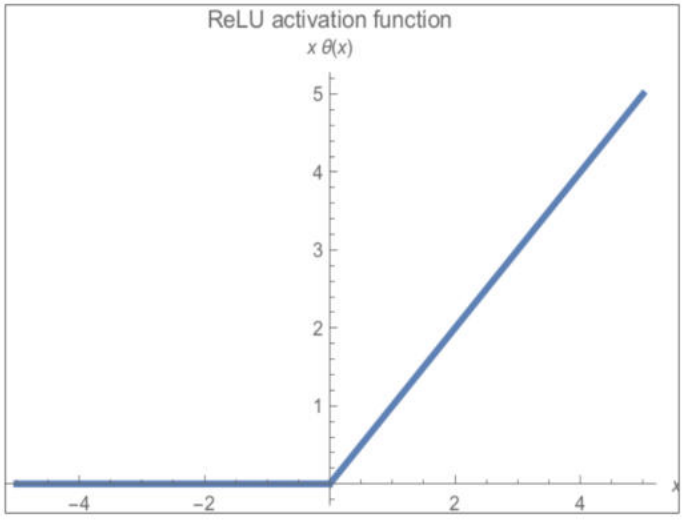

Figure 2.12 :ReLU plotted function[49]

b. Soft-max Function [63] [64]

The Soft-max function is a nonlinear activation function that is frequently used in the model's final layer to transform the network output into a probability distribution. [63].

When used in classification issues, especially multi-class classification problems, the Soft-max activation function is quite useful. Because the function's output has a probability distribution of 0 to 1, Soft-max provides the probabilities

of each category, with the target category having the greatest probability value. The Soft-max activation function was clarified by Equation (2.10), which was displayed in Figure (2.13).

$$\boldsymbol{\sigma}\,(\boldsymbol{Z})\boldsymbol{i} = \frac{e^{zi}}{\sum_{j=1}^{k} e^{zi}} \qquad 2.10$$

Where: **σ** is Soft-max , **Z** is input-vector , **K** is number of classes.

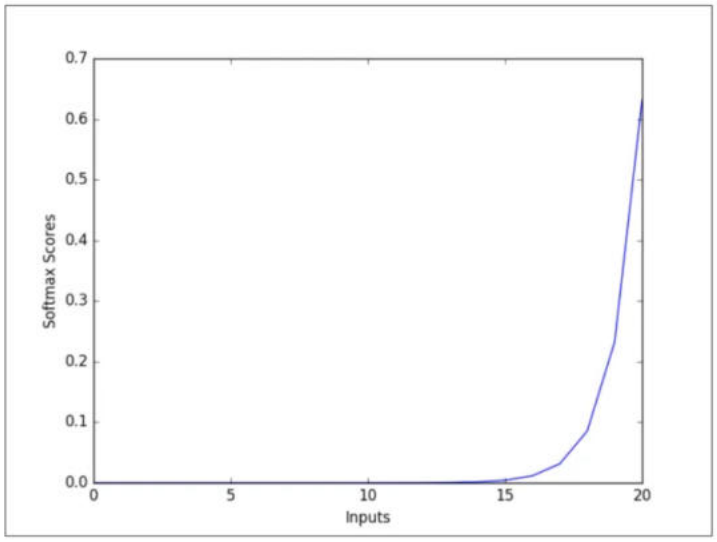

Figure 2.13 : Soft-max plotted function

2.8.6 Loss Function

The difference between the real output and the predicted output is used to calculate the error in early neural network models. Several formulae for calculating error in neural networks have surfaced recently; these formulas are known as Loss Functions [56]. Because different loss functions might result in different error values for the same prediction, the kind of loss function has a significant influence on the network's output. Loss functions can be divided into three categories: Classification Loss Functions, Regression Loss Functions, and Embedding Loss Functions are all examples of loss functions. With classification issues, classification loss functions are used. When the output variables are continuous,

the Regression Loss functions are used in regression problems. Embedding loss functions are used in jobs that require measuring the similarity of two inputs.

- **Categorical Cross-Entropy**

In multi-class classification tasks, this loss function is used. The target can only belong to one of several potential categories in these situations, and the model must choose which one. Because this function is utilized in networks that use the Soft-max activation function, it is frequently used to determine the difference between two probability distributions [65].

The following equation (2.11) illustrates the categorical cross-entropy.

$$\mathbf{L_{cross-entropy}}(\mathbf{y}.\acute{\mathbf{y}}) = -\sum_{\mathbf{i}} \mathbf{y_i} \log(\acute{\mathbf{y}}_{\mathbf{i}}) \qquad 2.11$$

Where $\boldsymbol{yi}$ actual target vector, $\tilde{\boldsymbol{y}}\boldsymbol{i}$ the output predicted vector, i vector length.

- **Mean Square Error Loss Function (MSE)**

The square difference between the real and anticipated value is measured by MSE, which is one of the most significant regression loss functions [66] . The mean square error loss is shown by the equation (2.12).

$$\boldsymbol{MSE} = \frac{\mathbf{1}}{\boldsymbol{n}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} (\boldsymbol{y_i} - \boldsymbol{p_i})^{\mathbf{2}} \qquad 2.12$$

Where $\boldsymbol{y_i}$ actual target vector, $\boldsymbol{p_i}$ the output predicted vector ,n vector length.

- **Euclidean Distance Loss**

This is the loss function. Embedding loss is most commonly employed in issues where two inputs must be compared, not in classification problems. It calculates the distance between two locations or vectors.

The Euclidean Distance Loss is shown in the following equation (2.13) [67].

$$\mathbf{Euclidean\ loss} = \sqrt{\sum_{\mathbf{i=1}}^{\mathbf{n}} (\mathbf{y_i} - \mathbf{p_i})^{\mathbf{2}}} \qquad 2.13$$

Where p_i is the predicted vector $f(y_i)$ is the actual input vector n is the length of the vector.

2.8.7 Type of optimization algorithms

One of the most crucial tasks is deciding the algorithm to use to improve a neural network. There are three types of optimization algorithms in machine learning. The first is batch or deterministic gradient techniques, which handle all training samples in a big batch at the same time. The second kind is stochastic or online techniques, which only employ one sample at a time. Most algorithms nowadays are a mix of the two. They only employ a portion of the training set at each epoch during training. Minibatch techniques are the name for these algorithms[56][54].

Two types of optimization methods can be distinguished: Algorithms with a constant learning rate, such as SGD, and Adaptive Learning Algorithms. The learning rate in the first group is set manually. In this sort of system, choosing the learning rate is a challenging problem. When a lower learning rate is used, the learning process is delayed, and the training time becomes excessive. When selecting a reasonably high learning rate, the loss value may fluctuate around the minimum amount, As a result, the convergence process is hampered. The algorithms in the second category, on the other hand, do not require human setting of the learning rate; instead, they use a heuristic method to change the learning rate value. In contrast to the algorithms in the first group, the algorithms in the second category have a variable learning rate during the training phase. As a result, numerous algorithms that fall within the two categories emerged (Stochastic Gradient Descent (SGD) and Adam) [54].

- **Adam**

Adam is an adaptive learning rate optimization method that assesses individual learning rates for a variety of variable [68]**.**

Adam automatically adjusts the learning parameter, which he does by estimating the first and second moments. But what is the current situation? The expectation of a random variable at the power of n is called the moment .The moment can illustrate in equation(2.14).

$$\boldsymbol{m_n = E[X^n]} \qquad 2.14$$

Where: **m** is the_moment, and **X** is a random_variable.

The following equations used to estimates, first and second moment Adam [69].

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \qquad 2.15$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \qquad 2.16$$

Where $\mathbf{m_t}$ and $\mathbf{v_t}$ are the preceding first and second moments, respectively, and they are both initialized to 0 in the first step. **β1,β2** are new parameters that have been added to the algorithm. The default values for these variables are 0.9 and 0.999, respectively.

Following the calculation of the first and second moments, the network weights are updated using the formulae below.(2.17).

$$w_t = w_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \qquad 2.17$$

Where W is network weights, η is Step size, $\epsilon = 10^{-8}$

2.9 Evaluation Metrics

When creating a machine learning model, evaluating performance and efficiency is crucial. In order for the machine learning model to be trustworthy, an assessment tool that is appropriate for the nature of the model's work must be chosen. When evaluating machine learning models, several scales are frequently employed to guarantee that the model is correctly evaluated. Machine learning evaluation measures are split into three categories: those used to assess classification tasks, those used to evaluate regression tasks, and those used to evaluate clustering tasks [70] [71] [72].

- **Evaluation Measures for Classification Tasks**

For **Classification** tasks, a variety of assessment metrics are provided, including Accuracy, Confusion Matrix, Recall, Precision, and F1 Score.

a) **Accuracy measure**

As indicated in equation, accuracy is defined as the ratio of the number of correct predictions to the total number of input samples (2.18) [72].

$$accuracy = \frac{number\ of\ correct\ prediction}{total\ number\ op\ prediction} \qquad 2.18$$

b) **Confusion Matrix (CM)**

One of the most significant tools for describing the classification model's performance is the classification model's performance matrix (CM). The confusion matrix for a binary classification model is shown in Figure (2.14) [73].

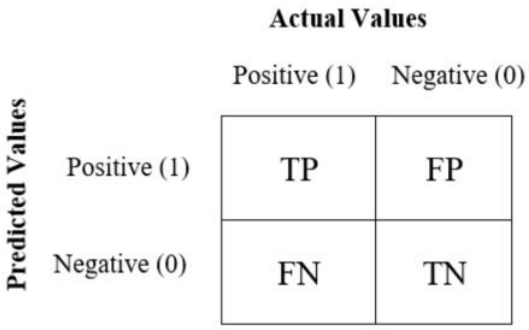

Figure 2.14: The confusion matrix

Each prediction will belong to one of the four categories:

- True positive (TP): correct positive prediction
- False positive (FP): incorrect positive prediction
- True negative (TN): correct negative prediction
- False negative (FN): incorrect negative prediction

The correctness of the matrix may be determined by calculating the average of the principal diagonal values using the equation (2.19).

$$\mathbf{acuraccy} = \frac{\mathbf{TP+TN}}{\mathbf{TP+TN+FN+FP}} \qquad 2.19$$

c) **Recall**

One of the most essential measures in models with imbalanced datasets is the recall. In the model, it calculates the genuine positive rate [72].

The following equation (2.20) may be used to determine this metric, which is dependent on the confusion matrix.

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \qquad 2.20$$

d) **Precision**

Precision, also known as positive predictive value, is calculated by dividing the total number of positive predictions by the number of actual positive forecasts [71]. The following equation (2.21) can be used to calculate precision metrics.

$$\boldsymbol{precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}} \qquad 2.21$$

e) **F1-score**

The F1-score may be thought of as a combined mean of recall and accuracy. The F1 attempts to strike a balance between recall and precision, and is used to assess a test's accuracy, which is determined by how many examples it properly classifies, as well as robustness, which is determined by the model's inability to ignore a large number of cases. The [0, 1] ranges in the F1 Score, with the higher value indicating better performance [73]. The F-score metric is represented by the equation (2.22) below.

$$\mathbf{F1score} = \frac{\mathbf{2*Precision*Recall}}{\mathbf{Precision+Recall}} \qquad 2.22$$

Chapter Three

THE PROPOSED SYSTEM METHODOLOGY

3.1 Introduction

This chapter describes the main stages of the practical side of the thesis and the methodology used to construct the proposed system. The proposed method involves several stages to be able to perform the task of classify email. The first section proposes an introduction to the chapter and the followed steps to get k-shingle technique, a hash of tokens, Min-hash technique done. The second section presents the classify email. Finally, the chapter highlights and the classify model.

3.2 Proposed Model

The suggested model's design is made up of many steps that work together to provide a model for classifying emails. The flow chart in Figure (3.1**)** depicts the system architecture as well as the general processes to categorize email. A dataset is an email that has been obtained from the internet and is ready to be utilized in this model.

To make the procedure easier in the next phases, the first stage marks removing the punctuation and the white space.

The k-shingle and subsequently Min-hash are used in the second stage to hash the dataset's contents after grouping it into tiny groups based on the number of words, which is k-shingle. The hash values for the shingle will be saved in a dictionary for further usage. This stage serves as a prelude to the subsequent phases of the procedure, which use (k-shingle) to check the hash value and compare it to the email. The Min-hash method is used in the third step to generate a new matrix with new values, such as the number of hashes and email address. The number of hashes provided to Min-hash represents the

number of hashes we want to compare, whereas the number of emails represents the number of emails we want to compare.

The fourth stage which includes split data for the training and testing sets.

The final stage is model extraction, where the neural network is trained to extract the classification system.

3.3 Data Set

The dataset used in this study can be found on kaggle, a machine learning database. There are 5725 "spam filter" instances in the dataset, with two columns, one for class and the other for the email string. Figure (3.2) shows a sample of the dataset.

	text	spam
0	Subject: naturally irresistible your corporate...	1
1	Subject: the stock trading gunslinger fanny i...	1
2	Subject: unbelievable new homes made easy im ...	1
3	Subject: 4 color printing special request add...	1
4	Subject: do not have money , get software cds ...	1
...	...	...
5721	Subject: re: research and development charges to...	0
5722	Subject: re: receipts from visit jim , thanks ...	0
5723	Subject: re: enron case study update wow ! all ...	0
5724	Subject: re: interest david , please , call sh...	0
5725	Subject: news : aurora 5 . 2 update aurora ve...	0

Figure 3.2: Dataset sample

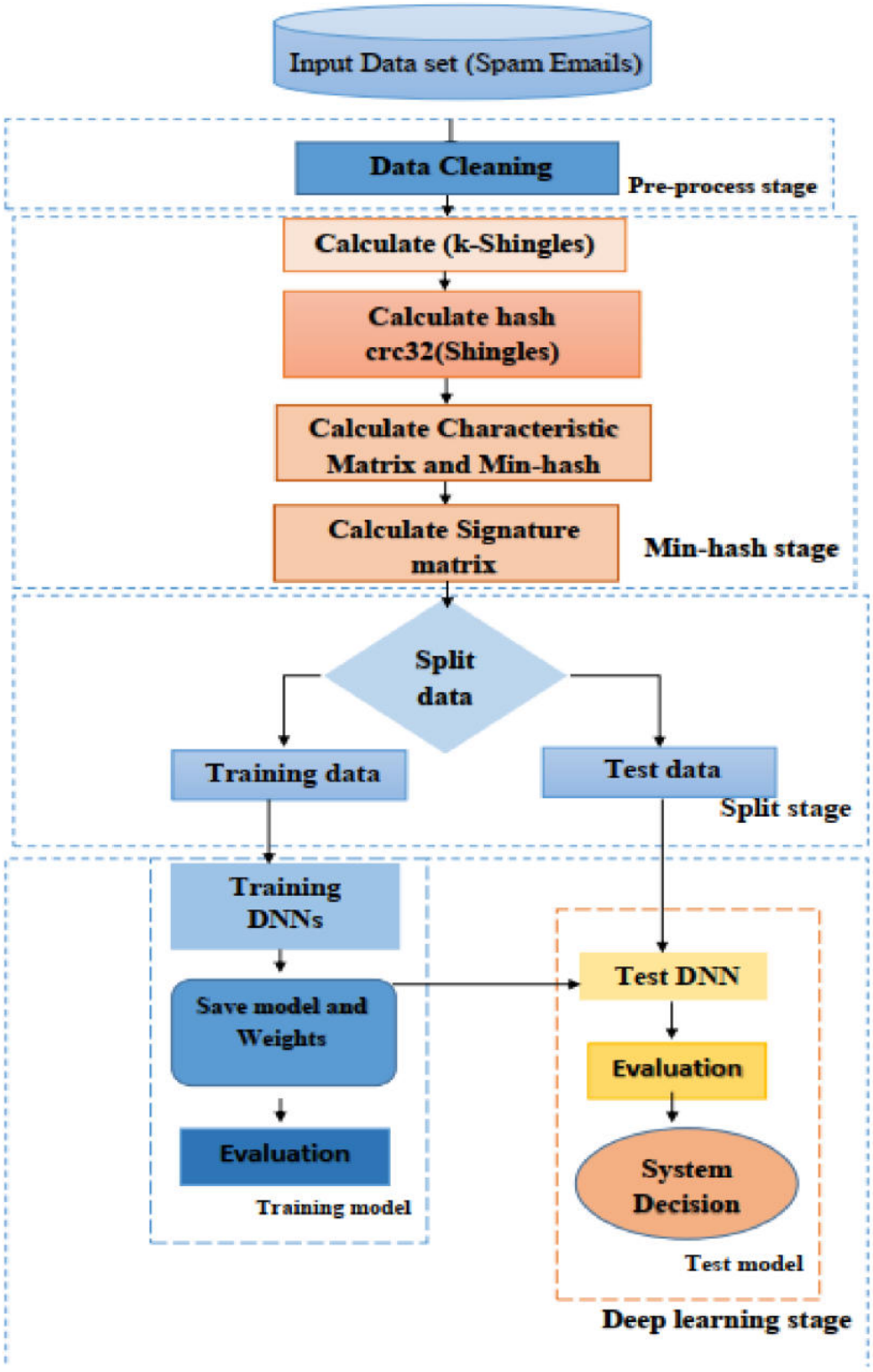

Figure 3.1: The proposed model of Min-hash deep learning

3.4 Preprocess Stage

Data cleaning is an important part of data science. Working with skewed data will result in a great number of issues. Breaking down sentences into words and dealing with punctuation and case are all part of this procedure. Unnecessary values, such as stop words, symbols, and punctuation marks, are eliminated, and data type conversion are in this work.

3.5 K-shingle Approaches

The k-shingle is a sequence of k tokens (characters) shown in an email. If the email is represented by a k-shingle. Hash shingles are occasionally helpful for slicing short phrases and expressing mails using groupings of hash values. As a result, the K-shingle functions in the following manner:

a. After Pre-process the email text by removing punctuation and adjusting the white space.

b. Choose the K words to divide the email into tokens.

c. Generate the characteristic matrix (S). If three email E1, E2, and E3 are used. The email E1 consists of "the sky is blue and rainy" with k=2. E2 consists of "the sky is blue" and E3 consists of "the sky".

Table (3.1) shows the characteristic matrix (S) that includes the tokens of shingles in column (1) and it is existing in email. The other two email E2 and E3 are in the same process.

Table 3.1: Characteristic matrix of sets based k-shingle

Shingles	E1	E2	E3
The sky	1	1	1

Sky is	1	1	0
Is blue	1	1	0
:	:	:	:
Shingle m	M	M	m

Algorithm (1): Shows the major steps of preprocess and k-shingle hashes for email.

Algorithm (1): K-shingle Algorithm	
Input	**Email E1, E2, E3.... En** **Number of K (shingle)**
Output	**Characteristic Matrix (M)**
Begin	
1	Preprocess the email text by - removing the punctuation - removing adjusting the white space
2	Set email to group based on k
3	Hashing set (shingling)
4	Find existing tokens in email
5	Generate characteristic Matrix
End	

3.6 Min-hash Technology

Min-hash is a technique that uses "random hash functions" to approximate the Jaccard Similarity between two distinct sets. The goal of Min-hash is to exchange large amounts of data using smaller representations known as signatures. The use of the Min-hash characteristic is critical since signatures are required to compare two sets of data and assess their similarity. As show in chapter 2.

3.6.1 Characteristic Matrices

The main goal of a search is to find relevant email matches. As a result, the project established a system for classifying emails as spam or ham for users security. Emails should be represented as groups before being matched for classify. This technique is known as shingling, and it essentially builds emails based on a set of N phrases to produce what it refers to as "Token". It is worth noting that there is a pre-processing phase before shingling that removes all spaces and extraneous punctuation marks, making the data clear enough to be organized into sets. For a phrase, shingles would be a collection of words produced from a sequence of words and then kept in a dictionary. For instance, if we have two shingles for the following statement “It is quite sunny today” the dictionary will look like {“it is quite”, “quite sunny”, “sunny today”}. The above example demonstrates what we did on this project to improve the search engine result.

After shingling email and converting it to sets, the method goes through a hashing procedure to condense the data and minimize its size. This phase will also identify the potential shingles email and convert it to K-shingles.

The email has been hashed and divided into smaller subsets formed from shingles. The creation of characteristic matrices is the next stage in the method process. It refers to the components of sets combined as rows, whereas it refers to each set or email as columns. When an element is part of the set, a cell receives a 1 value; otherwise, it receives a 0. The following Table (3.2) shows the way of representing shingles in the characteristic matrices.

Table 3.2: Example of Characteristic matrices

Shingles	E1	E2	E3	E4	H1	H2	…	Hn
B	0	0	1	0	3	0	…	12
E	0	0	1	0	15	14	…	3
A	1	0	0	1	10	11	…	9
D	1	0	1	1	5	10	…	12
C	0	1	0	1	0	6	…	1

The Min-hash hashing function is used for more accurate results, and it is another approach to express the characteristic matrix. The min-hash on an email is calculated by picking a modification of rows for a column of the previously described characteristic matrix. As it can be observed, each column's Min-hash amount is equal to the first row number in the permuted order where the column has value equal to 1. As a result, the signature matrix is made up of rows of hashes and columns of emails. Each column relates to the email's Min-hash signature. The signature matrix is significantly smaller than the characteristic matrix. The signature matrix is shown in the following table (3.3).

Table 3.3: The representation of Signature matrix

#	E1	E2	E3	E4
H1	5	0	3	0
H2	10	6	0	6

3.6.2 Generating a Min-hash Signature

Let say we have 10 random hash functions (10 different combinations of a and b).

a. Take the first hash function then applies it to all the shingle values in the email.

b. Find the Minimum hash (Minimum hash is the name of hashing algorithm) to use it as the first ingredient of the Min-hash signature.

c. Take the second hash and as before we should find the minimum hash value to use it as the second ingredient of the Min-hash signature and so on to the N values we have.

Algorithm (2) :Shows the main steps of Min-hash functions hashing on email.

Algorithm (2): Min-hash hashing functions	
Input	**- Characteristic Matrix M, Hash Functions h1, h2, h3, …, hn.** **- Picking n randomly hashing functions h1, h2, h3, …, hn.**

	- The signature Matrix S is constructed from characteristic Matrix M, where each row (i) is a hash function and each column (c) is a email. Then, set SIG (i,c) as signature matrix element for the hash h function and column c.
Output	**Signature Matrix (S)**
Begin	
1.	Convert the long bit vector into short signatures. To every column c in email, the next steps are done: • if c has 0 in both email rows r, do nothing. • if row has 1, then, for each i=1,2, ……, n set SIG(i,c) to the smaller value of the current value of SIG(i,c) and hi(r) Then Pr[hπ(c1) =hπ(c2)] =sim (c1, c2)
END	

3.7 Deep learning (DL)

Deep Learning is a type of machine learning that focuses on creating deep hierarchical data models. It theorizes that a hierarchy of intermediate representation is required to learn high-level data representations. Deep learning has been made possible by recent breakthroughs in learning algorithms for deep architectures, and deep learning systems have subsequently beaten or achieved state of the art performance on a variety of machine learning tasks.

A Neural Network is a parallel, distributed information processing structure. It is made up of processing units (each of which can have its own local memory and perform localized information processing operations) linked by unidirectional signal channels known as connections. Each processing element has a single output connection that branches out (or "fans out") into as many collateral connections as needed (each carrying the same signal - the processing element output signal). The output signal from the processing unit might be of any mathematical form. Each processing element's processing must be entirely local, relying solely on the current values of input signals coming via impinging connections and values stored in the processing element's local memory.

Back Propagation Neural Networks have a hierarchical design with completely linked layers or rows of processing units (with each unit itself comprised of several individual processing elements). Back Propagation belongs to the class of mapping Neural Network architectures.

Algorithm (3): Shows the Training Process of Back Propagation Algorithm.

Algorithm (3): Back Propagation Algorithm	
Input:	$f(x_i)$ // features vectors for four hash groups(H1,H2,H3,H4) in TRD
Output:	**Optimum weights**
Begin	
1	Initialize random weights and learning rate
2	The input unit receives h_i as input and sends it to the hidden unit
3	The net input of the **hidden** layer unit z_j is calculated

	as $$z_{0j} + \sum x_i * v_{ij}$$
4	Net Output of the **hidden** layer is calculated as $$z_j = f(z_{input})$$ the activation function is taken as binary or bipolar sigmoidal.
5	The net input of the **output** layer is calculated as $$z_j = f(z_{input})$$
6	The net output of the output layer: $\mathbf{f(y_{input})}$, the activation function is taken as binary or bipolar sigmoidal.
7	Calculation of error $\Delta e = (t_k - y_k) * \textbf{derivative of output layer } f(y_{input})$ where output unit y_k(k=1 to m) receives the target pattern corresponding to the input training pattern. Find out the derivative of the function.
8	Error correction and Weight Updating. **Weight Updating:** $\Delta w_{jk} = \alpha * \Delta e * z_j$, for k^{th} neuron $\Delta w_{0k} = \alpha * \Delta e$,for bias

	The error is sent backward.
9	The output units are updated: (y_k, k=1 to m) updates the bias and weights: $$\mathbf{w_{jk}(new) = w_{jk}(old) + \Delta w_{jk}}$$ $$\mathbf{w_{0k}(new) = w_{ok}(old) + \Delta w_{ok}}$$
10	Check for the stopping condition that is given as the number of epochs completed. The steps 2 to 9 are repeated until the stopping condition is obtained.
End	

- **Regularization of Deep Learning**

Regularization is a collection of strategies and methods for addressing the problem of over-fitting by lowering the generalization error while minimizing the training error. Overfitting is frequently caused by selecting excessively complicated models for the training data points. A simpler model, on the other hand, results in under-fitting the data. As a result, selecting the correct level of complexity in the model is crucial. Because the model's complexity cannot be deduced directly from the given training data, finding the correct model complexity for training is frequently impossible. This is when regularization kicks in, causing the complicated model to over-fit.

Dropout is a form of regularization that is commonly employed in neural networks. The remaining network is trained in the current iteration after connections between successive layers that are randomly deleted depending on

a dropout-ratio. Another set of random connections is discarded in the following cycle, as demonstrated in Chapter 2.

3.8 The Components of DNN

DNN consists of five fully connected layers (Dense layers) according to the following sizes in order (64,128,128,64,64,1), and 2 Dropout layers with size (0.1,0.2). The "ReLU" activation function is used in the dense layers. The first layer is the input layer, which gets the email feature ***f(xi),*** and the last layer is the output layer, which forecasts if the feature email spam or ham ***Pi*** is present. The remaining layers are the hidden layers responsible for detecting email features in the network. The architectures of the proposed DNN is shown in Figure (3.3), and the Table (3.4) shows the summary representation of each layer in DNN.

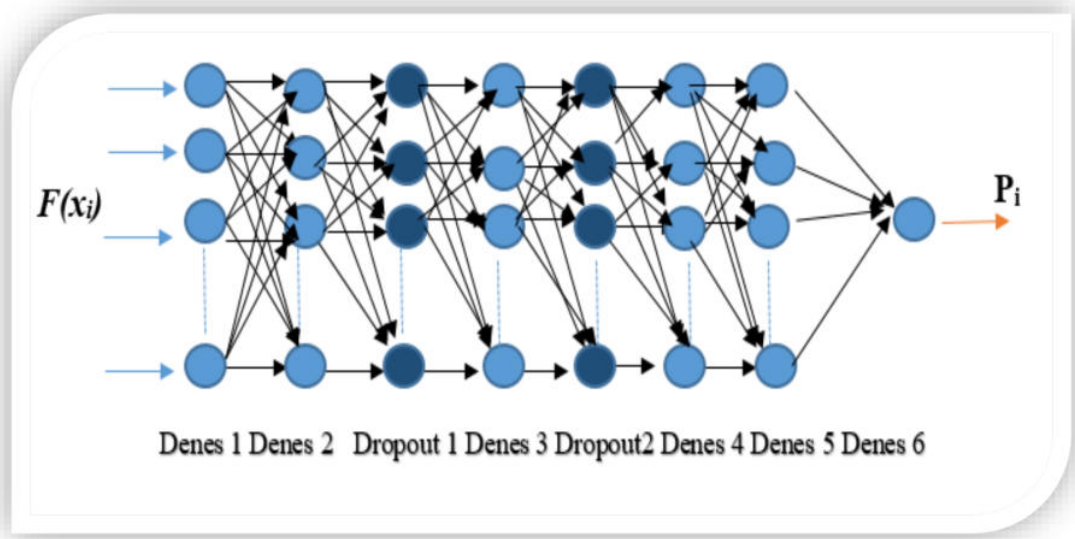

Figure 3.3: The architectures of proposed DNN for classify model

Table 3.4 :The summary represention of proposed DNN

Layer	Layer information	Output shape
Denes1 Activation	64 ReLU	320
Denes2 Activation	128 ReLU	8320
Dropout 1	0.1	0
Denes3 Activation	128 ReLU	16512
Dropout 2	0.2	0
Denes4 Activation	64 ReLU	8256
Denes5 Activation	64 ReLU	4160
Denes6 Activation	1 sigmoid	65

- **Training Model**

Deep Neural Network (DNN) in the model is trained the goal of training the network that is to find the spam email pattern for classify. To train network the training data must be provided. So, this model uses training dataset (TRD dataset) that sorted into four groups in the step of preparing data (Signature matrix).

- **The following steps explain the training mechanism of the multilayer algorithm**

- MLPs feed the data to the input layer of the network. Neuron layers connect in a graph so that the signal passes in one direction.
- MLPs compute the input with the weights that exist between the input layer and the hidden layers.
- MLPs use activation functions to determine which nodes to fire(active). Activation functions include ReLU, sigmoid, and tanh functions.
- MLPs train the model to understand the correlation and learn the dependencies between the independent and the target variables from a training data set.

Algorithm (4): Shows the deep neural network model training.

Algorithm (4) : DNN Model-Training	
Input:	$f(x_i)$ // features vectors for four hash groups(H1,H2,H3,H4) in TRD
Output:	Trained DNNs Model
Begin	
1	Let n number of Email groups in dataset
2	Call algorithm 1 // K-shingle Algorithm and pre

	processing
3	Call algorithm 2// Min-hash hashing functions
4	For i =1 to n do Call algorithm 3 // Create DNN (Create seven deep neural networks)
5	-Split data to TRD(70%) and TED (30%) -TRD split real training 70% and validation 30%
6	Train the four *DNN* on email groups H_I
7	Save weights *DNN*
End	

- **Testing Model**

Testing Phase of the model system is in the testing phase, the technical research, emails which are unlabeled emails as opposed to the emails in the training phase. The important properties of the testing phase of model system are made such that the system performs quickly to save time, take less space in memory, give better results with high accuracy. This testing phase can take only one unlabeled email for testing and return it to the correct class.

The testing phase of the model system consists of only classifying step because the obtained result are from the min-hash stage.

3.9 Evaluation Metrics

The last phase of this work is the four evaluation metrics that have been used for experimental results, as demonstrated in Chapter 2.

1- Accuracy measure

Classification when we use the term accuracy, we typically imply accuracy. The number of correct predictions divided by the total number of input samples is the ratio.

2- Recall

It is calculated by dividing the number of accurate positive findings by the total number of relevant samples (all samples that should have been identified as positive).

3- Precision

It is the number of correct positive outcomes divided by the classifier's expected number of positive findings**.**

4- F1-score

The Harmonic mean of accuracy and recall is the F1 Score. F1 Score has a range of [0, 1]. It informs you how exact and robust your classifier is (how many occasions it properly classifies). High precision but low recall offers an extremely accurate result, but it also misses a huge number of occurrences that are difficult to classify. The higher the F1 Score, the better our model's performance.

Chapter Four

Results and Discussion

4.1 Introduction

This chapter presents the experimental results and testing of each stage for the proposed system. It includes also a description of the dataset used with the proposed system, the hardware and software requirements in implementing the proposed system. The results of all stages are arranged based on their appearance in Chapter three.

4.2 System Requirement

The implementation of machine learning and deep learning algorithms require high computing power. Therefore, it is possible to obtain higher results in the case of implementation on higher specifications. The proposed system was implemented using the following hardware and software requirements showed in Table (4.1).

Table 4.1: Environment specifications for the proposed system.

• **Operating system**	Windows 10
• **Hardware**	
1. **Central Processing Unit (CPU)**	Intel(R) Core(TM) i5-7200U CPU.
2. **RAM**	8 GB.
3. **Hard Disk**	512 GB
• **Programming Language**	Python 3

The programming language Python version 3.8 was used, which is a high-level open source language that is easy to learn and adopts the object-oriented programming style. It is in constant updating, as the version is constantly updated, which requires constantly updating and modifying the code for the purpose of maintaining the results

4.3 Email Datasets

The dataset used in this study can be found in Kaggle web site, a machine learning database, as it has explained in the previous chapter in Section 3.3. Dataset is split into two subsets: training dataset (TRD) 70% and testing dataset (TED) 30%.

The TRD is also split into real training set 70%, and validation set 30%.

The ratio of division 70 to 30 was adopted because most of the research uses this ratio, and our experience is in different ratios, such as the ratio of 60 to 40, we obtained results in it for a small accuracy that did not exceed 88% at the various divisions of k-shingle, as well as the ratio of 80 to 20, the accuracy ratio did not exceed 70 % at different divisions of k-shingle .

4.4 Results of Data Preprocessing

The preprocessing stage that presented in section 3.4 is applied on email dataset before data enter to k shingle and min hash function. All emails that entered the proposed system must go through the pre-processing stage, whether it is in the training phase or the testing phase. The output of this stage is removing the punctuation, removing and adjusting the white space, as show in **Figure** (4.1 and 4.2).

Subject: naturally irresistible your corporate identity it is really hard to recollect a company: the market is full of suqgestions and the information is overwhelming; but a good catchy logo, # stylish statlonery and outstanding website will make the task much easier. we do not promise that having ordered a logo your company will automaticaily become a world leader: it is guide clear that without good products, effective business organization and practicable aim it will be hotat nowadays market; but we do promise that your marketing efforts will become much more effective. here is the list of clear benefits: creativeness: hand - made, original logos, specially done to reflect your distinctive company image. convenience: logo and stationery are provided in all formats; easy - to - use content management system lets you change your website content and

Figure 4.1: Dataset Before Preprocessing

'naturally', 'irresistible', 'your', 'corporate', 'identity', 'it', 'is', 'really', 'hard', 'to', 'recollect', 'a', 'company', 'the', 'market', 'is', 'full', 'of', 'suqgestions', 'and', 'the', 'information', 'isoverwhelminq', 'but', 'a', 'good', 'catchy', 'logo', 'stylish', 'statlonery', 'and', 'outstanding', 'website', 'will', 'make', 'the', 'task', 'much', 'easier', 'we', 'do', 'not', 'promise', 'that', 'having', 'ordered', 'a', 'logo', 'your', 'company', 'will', 'automaticaily', 'become', 'a', 'world', 'ieader', 'it', 'isguite', 'clear', 'that', 'without', 'good', 'products', 'effective', 'business', 'organization', 'and', 'practicable', 'aim', 'it', 'will', 'be', 'hotat', 'nowadays', 'market', 'but', 'we', 'do', 'promise', 'that', 'your', 'marketing', 'efforts', 'will', 'become', 'much', 'more', 'effective', 'here', 'is', 'the', 'list', 'of', 'clear', 'benefits', 'creativeness', 'hand', 'made', 'original', 'logos', 'specially', 'done', 'to', 'reflect', 'your', 'distinctive', 'company', 'image', 'convenience', 'logo', 'and', 'stationery', 'are', 'provided', 'in', 'all', 'formats', 'easy', 'to', 'use', 'content', 'management', 'system', 'lets', 'you', 'change', 'your', 'website', 'content', 'and', 'even', 'its', 'structure', 'promptness', 'you',

Figure 4.2: Dataset After Preprocessing

4.4.1 K-Shingle Results

In this section of testing, K-shingle approach is implemented as it has shown in chapter 3. Table (4.2) illustrates the result of implementing different length of k and different numbers. The k-shingle value considered is k=3 …, k=10.

Table 4.2: The results of Elapsed time, K-Shingle and next prime for dataset

Value of k	Number of Shingle(Tokens)	Time of divided to shingle in Min	Next prime
K=3	656780	28	656781
K=4	804340	36.6	804341
K=5	861087	37.8	861089
K=6	886636	40.1	886647
K=7	900729	40.7	900731
K=8	909729	41.5	909731
K=9	915912	42.9	915913
K=10	920037	42	920039

It is noticed that the number of shingle is increased when the number of emails increase too.

The following tables show examples of characteristic matrix of sets for each shingle, Table (4.3) when k=3, Table (4.4) when k=4 and Table (4.5) when k=5.

The existing of the shingle in email is representing by “1” and “0” if it does not exist in that email. Row “class" = "0" refers to that ham email and “1" to spam email.

Table 4.3: Example of Characteristic Matrix of Sets(K=3)

Shingles	E1	E2	E3	E4	E5	E6	E7	…	E5726
naturally irresistible your	1	0	0	0	0	0	0	…	0
irresistible your corporate	1	0	0	0	0	0	0	…	0
Your corporate identity	0	0	0	0	0	0	0	…	0
task much easier	1	1	0	1	0	0	1	…	0
logo your company	0	0	0	0	0	0	1	…	1
to reflect your	1	1	0	0	0	0	1	…	0
:	:	:	:	:	:	:	:	…	:
you will see	0	1	0	0	1	0	1	…	0
Class	1	1	0	1	1	1	0		0

Table 4.4: Example of Characteristic Matrix of Sets (K=4)

Shingles	E1	E2	E3	E4	E5	E6	E7	…	E5726
naturally irresistible your corporate	1	0	0	0	0	0	0	…	0
irresistible your corporate identity	1	0	0	0	0	0	0	…	0
Your corporate identity it	0	0	0	0	0	0	0	…	0
task much easier we	1	1	0	1	0	0	1	…	0
logo your company will	0	0	0	0	0	0	1	…	1
to reflect your distinctive	1	1	0	0	0	0	1	…	0
:	:	:	:	:	:	:	:	…	:
you will see logo	0	1	0	0	1	0	1	…	0
Class	1	1	0	1	1	1	0		0

Table 4.5: Example of Characteristic Matrix of Sets (K=5)

Shingles	E1	E2	E3	E4	E5	E6	E7	…	E5726
naturally irresistible your corporate identity	1	0	0	0	0	0	0	…	0

irresistible your corporate identity it	1	0	0	0	0	0	0	…	0
Your corporate identity it is	0	0	0	0	0	0	0	…	0
task much easier we do	1	1	0	1	0	0	1	…	0
logo your company will automaticaily	0	0	0	0	0	0	1	…	1
to reflect your distinctive company	1	1	0	0	0	0	1	…	0
:	:	:	:	:	:	:	:	…	:
you will see logo drafts	0	1	0	0	1	0	1	…	0
Class	1	1	0	1	1	1	0		0

The following tables show an example of generating hash for each shingle, table (4.6) when k=3, table (4.7) when k=4 and table (4.8) when k=5 .

Table 4.6: Example of Characteristics Matrix for Emails based on K-Shingles(k=3)

Hash values of shingles	E1	E2	E3	E4	E5	E6	E7	…	E5726
1011880877	1	0	0	0	0	0	0	…	0
0186651570	1	0	0	0	0	0	0	…	0

2556302256	0	0	0	0	0	0	0	...	0
0824185774	1	1	0	1	0	0	1	...	0
0785126324	0	0	0	0	0	0	1	...	1
1463686075	1	1	0	0	0	0	1	...	0
2508592058	0	0	0	0	0	0	0	...	1
3643544504	0	1	0	0	1	0	1	...	0
class	1	1	0	1	1	1	0		0

Table 4.7: Example of Characteristics Matrix for Emails based on K-Shingles(k=4)

Hash values of shingles	E1	E2	E3	E4	E5	E6	E7	...	E5726
0564272538	1	0	0	0	0	0	0	...	0
2774146459	1	0	0	0	0	0	0	...	0
2790661538	0	0	0	0	0	0	0	...	0
0817372587	1	1	0	1	0	0	1	...	0
2943753722	0	0	0	0	0	0	1	...	1
1396317696	1	1	0	0	0	0	1	...	0

:	:	:	:	:	:	:	:	...	:
3030523397	0	1	0	0	1	0	1	...	0
class	1	1	0	1	1	1	0		0

Table 4.8: Example of Characteristics Matrix for Emails based on K-Shingles(k=5**)**

Hash values of shingles	E1	E2	E3	E4	E5	E6	E7	...	E5726
4125629820	1	0	0	0	0	0	0	...	0
3792706933	1	0	0	0	0	0	0	...	0
2053905804	0	0	0	0	0	0	0	...	0
1621761431	1	1	0	1	0	0	1	...	0
4266794367	0	0	0	0	0	0	1	...	1
2858556807	1	1	0	0	0	0	1	...	0
:	:	:	:	:	:	:	:	...	:
934026617	0	1	0	0	1	0	1	...	0
class	1	1	0	1	1	1	0		0

The next step is finding the Min-hash based on the proposed approach, which is detailed in chapter 3.

4.4.2 K-shingle and Min-hash Results

As it has been proposed the characteristic matrix is fairly huge. Thus., it needs to be dense before using it. Therefore, Min-hash technique is used to generate signature matrix, as proposed in chapter 3. Tables (4.9, 4.10 and 4.11) explain the result of implementing hash function-based k-shingle to characteristic matrix. Each table refers to different k value as example we take (k=3, k=4, k=5).

Table 4.9: Values of Characteristic Matrix with Min-hash (k=3)

Hash values of k-shingles	E1	E2	E3	E4	E5	E6	E7	…	E5726	H1	H2	H3	H4
1011880877	1	0	0	0	0	0	0	…	0	1615	4686	2471	4885
186651570	1	0	0	0	0	0	0	…	0	1894	715	3273	4130
2556302256	0	0	0	0	0	0	0	…	0	2480	2059	144	11619
824185774	1	1	0	1	0	0	1	…	0	2434	9013	1068	64
785126324	0	0	0	0	0	0	1	…	1	5497	13395	691	24984
1463686075	1	1	0	0	0	0	1	…	0	4343	4234	4489	3982
2508592058	0	0	0	0	0	0	0	…	1	1412	4176	1031	5656
3643544504	0	1	0	0	1	0	1	…	0	4686	1180	146	48

Class	1	1	0	1	1	1	0	1	0				

Table 4.10: Values of Characteristic Matrix with Min-hash (k=4)

Hash values of k-shingles	E1	E2	E3	E4	E5	E6	E7	...	E5726	H1	H2	H3	H4
564272538	1		0	0	0	0	0	...	0	303	475	681	1287
2774146459	1	0	0	0	0	0	0	...	0	147	890	973	563
2790661538	0	0	0	0	0	0	0	...	0	380	859	144	149
817372587	1	1	0	1	0	0	1	...	0	347	376	68	498
2943753722	0	0	0	0	0	0	1	...	1	597	1265	6191	234
1396317696	1	1	0	0	0	0	1	...	0	498	694	789	282
:	0	0	0	0	0	0	0	...	1	:	:	:	:
3030523397	0	1	0	0	1	0	1	...	0	393	228	60	734
Class	1	1	0	1	1	1	0	1	0				

Table 4.11: Values of Characteristic Matrix with Min-hash (k=5)

Hash values of k-shingles	E1	E2	E3	E4	E5	E6	E7	...	E5726	H1	H2	H3	H4
4125629820	1	1	0	1	0	0	0	...	0	2854	30	3031	3979
3792706933	1	0	0	0	0	0	0	...	0	2979	2129	5160	26771
2053905804	0	0	0	0	0	0	0	...	0	15	486	77	18
1621761431	1	1	0	1	0	0	1	...	0	232	489	56	290
4266794367	0	0	0	0	0	0	1	...	1	301	362	310	1201
2858556807	1	1	0	0	0	0	1	...	0	221	686	529	2854
:	:	:	:	:	:	:	:	...	:	:	:	:	:
934026617	0	1	0	0	1	0	1	...	0	2437	1790	2339	208
Class	1	1	0	1	1	1	0	1	0				

Hence, from the characteristic matrix, a signature matrix is generated, and its size is less that characteristic matrix. Matrix's rows refer to the Min-hash number while its columns refer to the of email as it explained in Tables (4.12 , 4.13 and 4.14). We take example k=3 , k=4 ,k=5 as straight for explaining .

Table 4.12: Example of signature matrix of the email with hash functions (h=4& k=3**)**

hashes	E1	E2	E3	E4	E5	E6	E7	…	E5726
#1	3167	14225	1561	19421	7125	2265	2430	…	9950
#2	4214	2871	5590	1082	8825	391	1571	…	16055
#3	2471	5135	527	17070	4596	194	16514	…	19144
#4	1980	13730	892	64	48	2327	7277	…	24581
class	1	1	0	1	1	1	0	…	0

Table 4.13: Example of signature matrix of the email with hash functions (h=4& k=4)

hashes	E1	E2	E3	E4	E5	E6	E7	…	E5726
#1	2337	1443	1993	1210	1295	5634	1130	…	50
#2	1414	761	390	129	6775	391	752	…	155
#3	761	515	560	789	567	194	114	…	134
#4	990	313	119	124	48	2468	347	…	21
class	1	1	0	1	1	1	0	…	0

Table 4.14: Example of signature matrix of the email with hash functions (h=4& k=5)

hashes	E1	E2	E3	E4	E5	E6	E7	…	E5726
#1	995	2979	1827	1121	942	1872	119	…	3111
#2	818	2129	262	1628	704	1874	226	…	676
#3	901	5160	487	1432	229	2177	372	…	1213
#4	1494	26771	860	1242	220	351	432	…	45
class	1	1	0	1	1	1	0	…	0

Table 4.15 shows results of the elapsed time to found the signature matrix with k Different value .

Table 4.15 : Elapsed Time of signature matrix of the email

Value of k	Time of found signature matrix(min)
K=3	45.8
K=4	42.36
K=5	40.2
K=6	37.3
K=7	35.6
K=8	30.65

K=9	31.5
K=10	27

4.4.3 Deep Neural Network Results

In general, there are main parameters that control the training process of this model: batch size and the number of epochs. The following tables show the experimental results of the model for adjusting the values of these parameters during the training and validation phase.

Table (4.16) shows the process of adjusting batch size by comparison of the values of the loss function during training and validation when the batch size changes with the number of epochs.

Table (4.16) : The Batch size configuration with epochs =150 (k=3)

Batch size	Training Loss	Validation Loss
8	0.4	1.54
16	0.5	1.3
32	0.5	0.9
64	0.32	0.36

Table (4.17) shows the process of adjusting batch size by comparison of the values of the loss function during training and validation when the batch size changes with the number of epochs , value of k =4.

Table (4.17) :The Batch size configuration with epochs =150 (k=4)

Batch size	Training Loss	Validation Loss
8	0.37	0.38
16	0.34	0.35
32	0.32	0.33
64	0.31	0.32

Table (4.18) shows the process of adjusting batch size by comparison of the values of the loss function during training and validation when the batch size changes with the number of epochs ,value of k =5.

Table (4.18) :The Batch size configuration with epochs =150 (k=5)

Batch size	Training Loss	Validation Loss
8	0.3	0.31
16	0.3	0.37
32	0.25	0.307
64	0.21	0.29

Table (4.19) shows the results of model in training loss and validation loss when the number of epoch is different (k=3) .

Table 4.19: The experimental results of Model in training and validation sets ,k=3.

NO. of epoch	Training loss fun.	Validation loss fun.
10	1.16	1.8
20	1.4	1.45
40	1.1	1.4
70	0.4	0.6
90	0.39	0.4
120	0.37	0.38
150	0.32	0.36

Table (4.20) shows the results of model in training loss and validation loss when the number of epoch is different (k=4) .

Table 4.20: The experimental results of Model in training and validation sets ,k=4.

NO. of epoch	Training loss fun.	Validation loss fun.

10	1.2	1.7
20	0.6	0.7
40	0.5	0.6
70	0.49	0.46
90	0.43	0.45
120	0.32	0.33
150	0.31	0.32

Table (4.21) shows the results of model in training loss and validation loss when the number of epoch is different (k=5) .

Table 4.21: The experimental results of Model in training and validation sets ,k=5.

NO. of epoch	Training loss fun.	Validation loss fun.
10	1.40	1.5
20	0.75	0.6
40	0.5	0.7
70	0.43	0.6
90	0.34	0.38

120	0.3	0.33
150	0.21	0.29

4.5 Evaluate Results

The overall system performance is evaluated by evaluating the performance of the verification model based on the testing dataset TED. The metrics used to evaluate the verification model are CM, Accuracy, Precision, Recall, and F1-Measure as shown in chapter (3).

Table 4.22: The Performance Metrics of proposed system (k=3)

Metrics	Evaluate Results(%)
Accuracy	**90.02**
Precision	**89.0**
Recall	**89.0**
F1-Score	**89.0**

Table 4.23: The Performance Metrics of proposed system (k=4)

Metrics	Evaluate Results(%)
Accuracy	**89.1**
Precision	**87.0**
Recall	**87.0**

F1-Score	**88.0**

Table 4.24: The Performance Metrics of proposed system (k=5)

Metrics	Evaluate Results(%)
Accuracy	**98.5**
Precision	**97.0**
Recall	**95.0**
F1-Score	**98.0**

After finding the results for the models (k=3,k=4,k=5) and comparison among them, we find their accuracy to be (90,89,98.5) successively and we find the good accuracy when k =5.

However, to compare and discuss the performance of the proposed work with some of other existing works. In kaggle web site, some researchers present solutions for classification spam email ،and they find accuracy as shown in table (4.25).

Table 4.25: Comparison between verification accuracy of the proposed approach and other methods

Method	Accuracy (%)
Min-hash +DNN	98.5

Min-hash +Random Forest Classifier	86.0248
Nearest Neighbor Classifier	80.6848
Support Vector Classifier	82.4407
Adaboost Classifier	81.7384
Random Forest Classifier	85.0258
Gaussian Naive Bayes Classifier	46.2687
Decision Tree Classifier	77.5241
TFIDF + LSTM	76.0

The first two methods are the results of our search. The rest ones are the results from the work of researchers on web site (kaggel) . We note from the results in the table that the proposed method is superior to the other mentioned methods in terms of the searching goodness.

Chapter Five

Conclusions and Future Works

5.1 Conclusions

In this project, we used techniques K-shingle technique, Min-hash technique, deep neural network technique to classify emails. During the design and implementation of the proposed methodology, several notes can be concluded as an outcome of this study:

1- Min-hash technique has increase the quality of the whole work where it is used to find signature matrix. The findings show that the signature matrix is more sufficient for this mission, as it emphasizes tempo, secrecy, and honesty. The success criterion for consistency received a high ranking.

2- Throughout this work, a five-layer system of hidden layers was proposed. Starting with 64 nodes in the first secret layer and 128 nodes in the second, there were 128 nodes in the third layer, followed by 64 and 64 nodes in the fourth and fifth layers, respectively. 64 batches were set up after the training set was trained. The Python environment was used to carry out this work, and 70% to 30% of the data was taken for the purpose of training and testing the results.

3- Several ratios were used to divide the dataset we used 60% from data set to TRD and 40% from data set to TED, we obtained results in it for a small accuracy that did not exceed 88% at the various divisions of k - shingle. used 80% from data set to TRD and 20% from data set to TED, the accuracy ratio did not exceed 70 % at different divisions k-shingle. used 70% from data set to TRD and 30% from data set to TED, the accuracy90%when k=3 ,89% when k=4 and 98.5% when k=5 .

Thus, we conclude that the good accuracy was obtained when k=5.

4- The proposed model can be considered better than traditional methods in terms of accuracy and speed because it combines the characteristics of deep learning and data mining.

5.2 Future work

We summarize the future work in four points:

1. Implementing another algorithm such as frequency item set to generate attribute rules and company with min-hash data mining.
2. Implement the proposed work in a parallel open programing model such as map-reduce to reduce for time officiating.
3. Applying a clustering technique such as cure, chameleon and others instead of classification technique to find the methods of similarity.

REFRENCES

[1] O. Nasraoui, “Web data mining: exploring hyperlinks, contents, and usage data,” *Choice Reviews Online*, vol. 49, no. 05, pp. 49-2718-49–2718, 2012, doi: 10.5860/choice.49-2718.

[2] F. Sebastiani, “Machine learning in automated text categorization,” *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, Mar. 2002, doi: 10.1145/505282.505283.

[3] C. Häusler, “Method for Determining the Similarity of Documents,” pp. 1–13, 2012.

[4] M. Student, “email Spam Detection using Naive Bayes Classifier,” *International Journal of Computer Sciences and Engineering*, vol. 4, no. 6, pp. 934–938, Jul. 2018, doi: 10.26438/ijcse/v6i7.934938.

[5] M. A. Hassan and N. Mtetwa, “Feature Extraction and Classification of Spam Emails,” in *2018 5th International Conference on Soft Computing & Machine Intelligence (ISCMI)*, Nov. 2018, pp. 93–98, doi: 10.1109/ISCMI.2018.8703222.

[6] S. Gric and Z. Hutinski, “INFORMATION SYSTEM SECURITY THREATS CLASSIFICATIONS,” *Journal of information and organizational sciences*, vol. 31, pp. 51–61, 2007.

[7] M. V. Kulkarni, M. S. Patil, C. H. Patil, and M. S. Kirdat, “Study on Network Security Algorithm,” *International Journal of Engineering Research and Technology (IJERT)*, vol. 8, no. 5, pp. 1–3, 2020, [Online]. Available: www.ijert.org.

[8] T. A. Eze and C. C.-E. Aroh, "Management of Information Security in Public Universities in Nigeria," *International Journal for Digital Society*, vol. 11, no. 1, pp. 1575–1578, Mar. 2020, doi: 10.20533/ijds.2040.2570.2020.0196.

[9] R. von Solms and J. van Niekerk, "From information security to cyber security," *Computers & Security*, vol. 38, pp. 97–102, Oct. 2013, doi: 10.1016/j.cose.2013.04.004.

[10] T. Ayodele, S. Zhou, and R. Khusainov, "Email Classification Using Back Propagation Technique," *International Journal of Intelligent Computing Research (IJICR)*, vol. 1, no. 1, pp. 3–9, Mar. 2010, doi: 10.20533/ijicr.2042.4655.2010.0001.

[11] R. D. Lakshmi and N. Radha, "Supervised Learning Approach for Spam Classification Analysis using Data Mining Tools," *International Journal on Computer Science and Engineering (IJCSE)*, vol. 2, no. 8, pp. 2760–2766, 2010, [Online]. Available: http://infonomics-society.org/wp-content/uploads/ijicr/published-papers/volume-1-2010/Email-Classification-Using-Back-Propagation-Technique.pdf.

[12] arun k Pujari, *Data Mining by A K Pujari.pdf*. india, 2008.

[13] D. K. Dewangan and P. Gupta, "Email Spam Classification Using Support Vector Machine Algorithm.pdf," *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. 6, no. VI, June 2018-Available at www.ijraset.com. 2018.

[14] Chih-Chin Lai and Ming-Chi Tsai, "An Empirical Performance Comparison of Machine Learning Methods for Spam E-Mail Categorization," in *Fourth International Conference on Hybrid Intelligent Systems (HIS'04)*, 2005, pp.

44–48, doi: 10.1109/ICHIS.2004.21.

[15] S. Youn and D. McLeod, “Spam email classification using an adaptive ontology,” *Journal of Software*, vol. 2, no. 3. pp. 43–55, 2007, doi: 10.4304/jsw.2.3.43-55.

[16] B. Yu and Z. ben Xu, “A comparative study for content-based dynamic spam classification using four machine learning algorithms,” *Knowledge-Based Systems*, vol. 21, no. 4, pp. 355–362, 2008, doi: 10.1016/j.knosys.2008.01.001.

[17] A. Kumar and S. Sahni, “A Comparative Study of Classification Algorithms for Spam Email Data Analysis,” *International Journal on Computer Science and Engineering (IJCSE)*, vol. 3, no. 5, pp. 1890–1895, 2011, [Online]. Available: http://www1.ics.uci.edu/~mlearn/MLRepository.html.

[18] D. DeBarr and H. Wechsler, “Spam detection using Random Boost,” *Pattern Recognition Letters*, vol. 33, no. 10, pp. 1237–1244, Jul. 2012, doi: 10.1016/j.patrec.2012.03.012.

[19] C. C. Aggarwal and C. Zhai, “A Survey of Text Classification Algorithms,” in *Mining Text Data*, vol. 9781461432, Boston, MA: Springer US, 2012, pp. 163–222.

[20] S. Seth and S. Biswas, “Multimodal Spam Classification Using Deep Learning Techniques,” in *2017 13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, Dec. 2017, vol. 978-1–5386, pp. 346–349, doi: 10.1109/SITIS.2017.91.

[21] A. Alghoul, S. Al Ajrami, G. Al Jarousha, G. Harb, Abu-Naser, and S. S.,

"Email Classification Using Artificial Neural Network," *International Journal of Academic Engineering Research (IJAER)*, vol. 2, no. 11, pp. 8–14, 2018, [Online]. Available: https://ieeexplore.ieee.org/document/8707394/.

[22] M. Bassiouni, M. Ali, and E. A. El-Dahshan, "Ham and Spam E-Mails Classification Using Machine Learning Techniques," *Journal of Applied Security Research*, vol. 13, no. 3, pp. 315–331, Jul. 2018, doi: 10.1080/19361610.2018.1463136.

[23] S. Sumathi and G. K. Pugalendhi, "Cognition based spam mail text analysis using combined approach of deep neural network classifier and random forest," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 6, pp. 5721–5731, Jun. 2020, doi: 10.1007/s12652-020-02087-8.

[24] I. Alsmadi and I. Alhami, "Clustering and classification of email contents," *Journal of King Saud University - Computer and Information Sciences*, vol. 27, no. 1, pp. 46–57, 2015, doi: 10.1016/j.jksuci.2014.03.014.

[25] D. L. Olson and D. Delen, *Advanced Data Mining Techniques*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[26] D. Coss and S. Samonas, "The CIA Strikes Back: Redefining Confidentiality, Integrity and Availability in Security.," *Journal of Information System Security*, vol. 10, no. 3, pp. 21–45, 2014.

[27] A. F. Sheikh, *CompTIA Security+ Certification Study Guide*. Berkeley, CA: Apress, 2020.

[28] M. Warkentin and C. Orgeron, "Using the security triad to assess blockchain technology in public sector applications," *International Journal of*

Information Management, vol. 52, p. 102090, Jun. 2020, doi: 10.1016/j.ijinfomgt.2020.102090.

[29] T. Fawcett, "'In vivo' spam filtering: A challenge problem for data mining," May 2004, [Online]. Available: http://arxiv.org/abs/cs/0405007.

[30] G. Jain, M. Sharma, and B. Agarwal, "Spam detection in social media using convolutional and long short term memory neural network," *Annals of Mathematics and Artificial Intelligence*, vol. 85, no. 1, pp. 21–44, 2019, doi: 10.1007/s10472-018-9612-z.

[31] A. Dubey, S. Chakrabarti, and C. Bhattacharyya, "Diversity in ranking via resistive graph centers," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2011, pp. 78–86, doi: 10.1145/2020408.2020428.

[32] S. Agarwal, "Data mining: Data mining concepts and techniques," *Proceedings - 2013 International Conference on Machine Intelligence Research and Advancement, ICMIRA 2013*. pp. 203–207, 2014, doi: 10.1109/ICMIRA.2013.45.

[33] E. H. Herskovits and J. P. Gerring, "Application of a data-mining method based on Bayesian networks to lesion-deficit analysis," *NeuroImage*, vol. 19, no. 4, pp. 1664–1673, 2003, doi: 10.1016/S1053-8119(03)00231-3.

[34] M. J. Zaki and M. J. Meira, "Data Mining and Analysis: Fundamental Concepts and Algorithms." p. 562, 2013, [Online]. Available: https://books.google.com.tr/books?id=Gh9GAwAAQBAJ&lpg=PR9&dq=Data Mining and Analysis: Foundations and Algorithms&hl=tr&pg=PR9#v=onepage&q=Data Mining and Analysis:

Foundations and Algorithms&f=false.

[35] R. J. Lewis, D. Ph, and W. C. Street, “An Introduction to Classification and Regression Tree (CART) Analysis,” *2000 Annual Meeting of the Society for Academic Emergency Medicine*, no. 310, p. 14p, 2000, [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.95.4103&rep=rep1&type=pdf.

[36] I. Mokriš and L. Skovajsová, “Comparison of two document clustering techniques which use neural networks,” in *ICCC 2008 - IEEE 6th International Conference on Computational Cybernetics, Proceedings*, 2008, vol. 400, pp. 75–78, doi: 10.1109/ICCCYB.2008.4721382.

[37] O. Ertl, “SuperMinHash - A New Minwise Hashing Algorithm for Jaccard Similarity Estimation,” no. 3, Jun. 2017, [Online]. Available: http://arxiv.org/abs/1706.05698.

[38] T. Euclidean, “4 Jaccard Similarity and Shingling,” pp. 2–5, [Online]. Available: https://www.cs.utah.edu/~jeffp/teaching/cs5955/L4-Jaccard+Shingle.pdf.

[39] F. Deng, S. Siersdorfer, and S. Zerr, “Efficient jaccard-based diversity analysis of large document collections,” in *Proceedings of the 21st ACM international conference on Information and knowledge management - CIKM ’12*, 2012, p. 1402, doi: 10.1145/2396761.2398445.

[40] T. Christiani and R. Pagh, “Set similarity search beyond MinHash,” in *Proceedings of the Annual ACM Symposium on Theory of Computing*, 2017, vol. Part F1284, pp. 1094–1107, doi: 10.1145/3055399.3055443.

[41] B. F. Momin, P. J. Kulkarni, and A. Chaudhari, "Web document clustering using document index graph," *Proceedings - 2006 14th International Conference on Advanced Computing and Communications, ADCOM 2006*, pp. 32–37, 2006, doi: 10.1109/ADCOM.2006.4289851.

[42] M. E. Manaa and G. Abdulameer, "Web documents similarity using K-shingle tokens and minHash technique," *Journal of Engineering and Applied Sciences*, vol. 13, no. 6, pp. 1499–1505, 2018, doi: 10.3923/jeasci.2018.1499.1505.

[43] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep Learning for Computer Vision: A Brief Review," *Computational Intelligence and Neuroscience*, vol. 2018, 2018, doi: 10.1155/2018/7068349.

[44] S. C. Peter *et al.*, "Encyclopedia of Bioinformatics and Computational Biology:," vol. 1–3, no. January, pp. 661–676, 2018.

[45] H. Kaur and E. P. Verma, "Survey on E-Mail Spam Detection Using Supervised Approach," *International Journal of Engineering Sciences & Research Technology Ijesrt*, vol. 6, no. 4. pp. 120–128, 2017, doi: 10.5281/zenodo.496096.

[46] D. B. Batchelder, "Living wet: Baptismal remembrance and life at the edge of dawn," *Liturgy*, vol. 21, no. 4. pp. 11–17, 2006, doi: 10.1080/04580630600872547.

[47] R. E. Neapolitan and X. Jiang, *Neural Networks and Deep Learning*. 2018.

[48] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep Models under the GAN: Information leakage from collaborative deep learning," *Proceedings of the*

ACM Conference on Computer and Communications Security, pp. 603–618, 2017, doi: 10.1145/3133956.3134012.

[49] J. Schmidhuber, "Deep Learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015, doi: 10.1016/j.neunet.2014.09.003.

[50] A. Zamir, H. U. Khan, W. Mehmood, T. Iqbal, and A. U. Akram, "A feature-centric spam email detection model using diverse supervised machine learning algorithms," *Electronic Library*, vol. 38, no. 3. pp. 633–657, 2020, doi: 10.1108/EL-07-2019-0181.

[51] R. Vinayakumar, H. B. Barathi Ganesh, M. Anand Kumar, K. P. Soman, and P. Poornachandran, "DeepAnti-PhishNet: Applying deep neural networks for phishing email detection CEN-AISecurity@IWSPA-2018," *CEUR Workshop Proceedings*, vol. 2124. pp. 39–49, 2018.

[52] A. Rajendra Kurup, M. Ajith, and M. Martínez Ramón, "Semi-supervised facial expression recognition using reduced spatial features and Deep Belief Networks," *Neurocomputing*, vol. 367, pp. 188–197, 2019, doi: 10.1016/j.neucom.2019.08.029.

[53] B. Mehlig, *Machine learning with neural networks An introduction for scientists and engineers*. 2019.

[54] H. Wang, R. Czerminski, and A. C. Jamieson, "Neural Networks and Deep Learning," in *The Machine Age of Customer Insight*, 2021, pp. 91–101.

[55] P. Fuleky, *Macroeconomic Forecasting in the Era of Big Data*, vol. 52, no. June. 2020.

[56] R. Ramachandran, D. C. Rajeev, S. G. Krishnan, and P. Subathra, "Deep

learning in neural networks," *International Journal of Applied Engineering Research*, vol. 10, no. 10. pp. 25433–25448, 2015.

[57] X. Li, Y. Xu, Q. Lv, and Y. Dou, "Affine-transformation parameters regression for face alignment," *IEEE Signal Processing Letters*, vol. 23, no. 1, pp. 55–59, 2016, doi: 10.1109/LSP.2015.2499778.

[58] R. K. B, M. S. Pini, and F. Rossi, "Spam Filtering Using Regularized Neural Networks with Rectified Linear Units," *Springer International Publishing AG 2016*, vol. 10037, pp. 250–265, 2016, doi: 10.1007/978-3-319-49130-1.

[59] Y. Bengio, *Learning deep architectures for AI*, vol. 2, no. 1. 2009.

[60] S. Raschka, "Mlxtend 0.9.0," pp. 1–388, 2018, [Online]. Available: https://rasbt.github.io/mlxtend/.

[61] I. G. and Y. B. and A. Courville, "Deep learning Book pdf," *Nature*, vol. 29, no. 7553. pp. 1–73, 2016.

[62] Y. Pan *et al.*, "Brain tumor grading based on Neural Networks and Convolutional Neural Networks," *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, vol. 2015-Novem, pp. 699–702, 2015, doi: 10.1109/EMBC.2015.7318458.

[63] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5455–5516, 2020, doi: 10.1007/s10462-020-09825-6.

[64] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "A survey on deep learning for big

data," *Information Fusion*, vol. 42, no. November 2017, pp. 146–157, 2018, doi: 10.1016/j.inffus.2017.10.006.

[65] K. Janocha and W. M. Czarnecki, "On loss functions for deep neural networks in classification," *Schedae Informaticae*, vol. 25. pp. 49–59, 2016, doi: 10.4467/20838476SI.16.004.6185.

[66] J. Qi, J. Du, S. M. Siniscalchi, X. Ma, and C. H. Lee, "On Mean Absolute Error for Deep Neural Network Based Vector-to-Vector Regression," *IEEE Signal Processing Letters*, vol. 27, no. c, pp. 1485–1489, 2020, doi: 10.1109/LSP.2020.3016837.

[67] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa, "An All-In-One Convolutional Neural Network for Face Analysis," *IEEE International*, pp. 17–24, 2017, doi: 10.1109/FG.2017.137.

[68] S. Ruder, "An overview of gradient descent optimization algorithms," pp. 1–14, 2016, [Online]. Available: http://arxiv.org/abs/1609.04747.

[69] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–15, 2015.

[70] C. Ferri, J. Hernández-Orallo, and R. Modroiu, "An experimental comparison of performance measures for classification," *Pattern Recognition Letters*, vol. 30, no. 1, pp. 27–38, 2009, doi: 10.1016/j.patrec.2008.08.010.

[71] Q. Gu, L. Zhu, and Z. Cai, "Evaluation measures of the classification performance of imbalanced data sets," *Communications in Computer and Information Science*, vol. 51, pp. 461–471, 2009, doi: 10.1007/978-3-642-

04962-0_53.

[72] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” *Information Processing and Management*, vol. 45, no. 4, pp. 427–437, 2009, doi: 10.1016/j.ipm.2009.03.002.

[73] J. Novakovic, A. Veljovi, S. Iiic, Z. Papic, and M. Tomovic, “Evaluation of Classification Models in Machine Learning,” *Theory and Applications of Mathematics & Computer Science*, vol. 7, no. 1, pp. 39–46, 2017, [Online]. Available: https://uav.ro/applications/se/journal/index.php/TAMCS/article/view/158.

الخلاصة

أصبح الإنترنت جزءًا لا يتجزأ من الحياة الحديثة، اذ يعد التواصل من أهم جوانبه. يعرف البريد الإلكتروني على انه أداة اتصال يمكن استخدامها للأغراض الشخصية والمهنية (الرسمية). ان رسائل السبام (spam messages) هي رسائل تلقى من قبل المستخدمين دون قصد أو بشكل عشوائي ولذلك تعتبر هذه الرسائل غير مرغوب بها. يتم استخدام البريد الالكتروني من قبل مجموعة كبية من الاشخاص وبشكل يومي للتواصل على مستوى العالم. في الوقت الحالي، هنالك عدد كبير من رساءل البريد الالكتروني العشوائية والتي تعتبر حينات منطقية. ان كثرة تواجد هذه الرسائل العشوائية غير المرغوب بها يسبب أشكالا كبيرا لكل من مستخدمي الانترنت والاطراف التي تقدم خدماتها على حد سواء. من المشاكل التي تسببها الرسائل العشوائية غير المرغوب بها هي تدهور بيانات تحليل المستخدم ، وتشجيع ترحيل فيروسات الشبكة ، وتوسيع حركة الترتيب المكدس، اشغال سيرفر تخزين البريد ، بالاضافة الى استهلاك الوقت وعرض النطاق الترددي للشبكة واستنفاذ حيوية رسائل البريد الإلكتروني الحقيقية بين البريد العشوائي.

لذلك من الضروري منع انتشار البريد الإلكتروني العشوائي. نظرًا لوجود العديد من تقنيات التنقيب عن البيانات المفيدة في الحفاظ على الأمان ، يمكن أيضًا أن تكون مفيدة في تصنيف البريد الإلكتروني العشوائي. بالنسبة إلى العمل الحالي ، يتم دمج تقنية min-hash مع خوارزمية Deep Neural Network (DNN) لتصنيف رسائل البريد الإلكتروني إلى رسائل عشوائة (spam) ورسائل اعتيادية (ham) . يعتمد Min-hash بشكل أساسي على مفهومين رئيسيين هما تشابه Jaccard وK-shingle ، حيث تم استخدام عدة امثلة لتقسيم ال K-shingle مثل (k=3, 4, 5) عند مقارنة النتائج.

من خلال هذه الدراسة ، تم اقتراح نظام يتكون من خمس طبقات مخفية، بدءًا من ٦٤ عقدة في الطبقة السرية الأولى و ١٢٨ عقدة في الثانية ، كان هناك ١٢٨ عقدة في الطبقة الثالثة ، تليها ٦٤ و ٦٤ عقدة في الطبقتين الرابعة والخامسة على التوالي. تم إعداد ٦٤ دفعة بعد تدريب مجموعة التدريب. تم استخدام بيئة Python لتنفيذ هذا العمل ، وتم أخذ ٧٠٪ إلى ٣٠٪ من البيانات لغرض التدريب واختبار النتائج. أظهرت النتائج أنه تم الحصول على معدل دقة مرتفع بشكل ملحوظ (٩٨,٥٪) باستخدام التوليفة (k = 5) ، مما يعني أنها طريقة فعالة ويجب اعتمادها وتطويرها بشكل أكبر في مجال الكشف عن الرسائل الاقتحامية وتصنيفها.

وزارة التعليم العالي والبحث العلمي
جامعة بابل
كلية العلوم للبنات
قسم علوم الحاسوب

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة بابل

تصنيف رسائل البريد الإلكتروني (Spam)باستخدام Min-Hash والتعلم العميق

رسالة
مقدمة الى مجلس كلية العلوم للبنات – جامعة بابل كجزء من
متطلبات نيل درجة الماجستير في العلوم/علوم الحاسوب

من قبل الطالبة
نهى حسين مرزة حمزة

بإشراف
أ. د. حسين عطية لفته
أ.م.د. مهدي عبادي مانع

٢٠٢٢ م ١٤٤٣ هـ